



0130/  
2131  
PATENT  
Attorney Docket No.: 678-812 (P10046)

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

**APPLICANTS:** Sung-Ho CHOI

**SERIAL NO.:** 10/085,776

**FILED:** February 27, 2001

**DATED:** April 8, 2002

**FOR:** APPARATUS AND METHOD FOR CODING/DECODING  
TFCI BITS IN AN ASYNCHRONOUS CDMA COMMUNICATION  
SYSTEM

**RECEIVED**

APR 30 2002

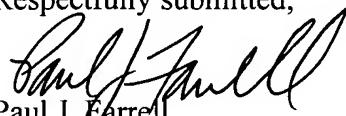
Technology Center 2100

**TRANSMITTAL OF CERTIFIED COPY**

Sir:

Enclosed is a certified copy of Korean Patent Application No. 10150 filed on February 27, 2001 and from which priority is claimed under 35 U.S.C. § 119.

Respectfully submitted,

  
Paul J. Farrell

Reg. No. 33,494

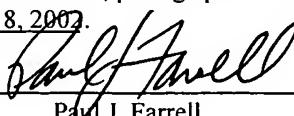
Attorney for Applicant(s)

**DILWORTH & BARRESE, LLP**  
**333 Earle Ovington Blvd.**  
**Uniondale, NY 11553**  
**TEL: (516) 228-8484**  
**FAX: (516) 228-8516**  
**PJF/lah**

**CERTIFICATION UNDER 37 C.F.R. § 1.8(a)**

I hereby certify that this correspondence (and any document referred to as being attached or enclosed) is being deposited with the United States Postal Service as first class mail, postage paid in an envelope addressed to: Commissioner for Patents, Washington, D.C. 20231 on April 8, 2002.

Dated: April 8, 2002

  
Paul J. Farrell

Sung-Ho Choi, et al.  
S.N. 101085,776  
ATTY. DOC.: 678-812(P10046)



RECEIVED

APR 30 2002

Technology Center 2100

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

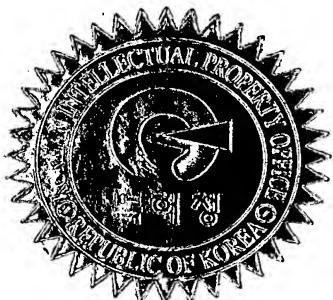
This is to certify that the following application annexed hereto  
is a true copy from the records of the Korean Intellectual  
Property Office.

PRIORITY DOCUMENT  
CERTIFIED COPY OF

출 원 번 호 : 특허출원 2001년 제 10150 호  
Application Number PATENT-2001-0010150

출 원 년 월 일 : 2001년 02월 27일  
Date of Application FEB 27, 2001

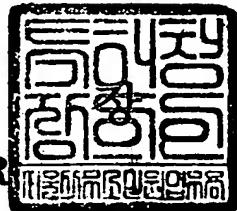
출 원 인 : 삼성전자 주식회사  
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.



2002 년 02 월 27 일

특 허 청

COMMISSIONER



## 【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0020
【제출일자】	2001.02.27
【국제특허분류】	H04B
【발명의 명칭】	분할모드에 따른 전송형식 조합표시 비트의 전송 장치 및 방법 apparatus and method for transmitting TFCI bits in cdma communication system
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	이건주
【대리인코드】	9-1998-000339-8
【포괄위임등록번호】	1999-006038-0
【발명자】	
【성명의 국문표기】	최성호
【성명의 영문표기】	CHOI, Sung-Ho
【주민등록번호】	700405-1268621
【우편번호】	463-010
【주소】	경기도 성남시 분당구 정자동 느티마을 305동 302호
【국적】	KR
【발명자】	
【성명의 국문표기】	김재열
【성명의 영문표기】	KIM, Jae Yoel
【주민등록번호】	700219-1047637
【우편번호】	435-042
【주소】	경기도 군포시 산본2동 백두아파트 960동 1401호
【국적】	KR

1020010010150

출력 일자: 2002/2/28

【발명자】

【성명의 국문표기】 이현우

【성명의 영문표기】 LEE,Hyun Woo

【주민등록번호】 630220-1709811

【우편번호】 441-390

【주소】 경기도 수원시 권선구 권선동 벽산 아파트 806동  
901호

【국적】 KR

【취지】 특허법 제42조의 규정에 의하여 위와 같이 출원합니다. 대리인  
이건주 (인)

【수수료】

【기본출원료】 20 면 29,000 원

【가산출원료】 45 면 45,000 원

【우선권주장료】 0 건 0 원

【심사청구료】 0 항 0 원

【합계】 74,000 원

【첨부서류】 1. 요약서·명세서(도면)\_1통

**【요약서】****【요약】**

비동기 이동통신 시스템의 DSCH채널과 DCH채널의 TFCI를 전송하는 방법이,  
SRNC가 DSCH 데이터를 DRNC에 전송하며, 상기 DRNC가 상기 DSCH 데이터의 전송시  
간을 결정한 후 해당하는 TFCI를 SRNC에 전송하고 상기 DSCH 데이터를 노드 B에  
DSCH 데이터를 전송하는 과정과, 상기 SRNC가 DCH 데이터 및 TFCI를 결정하여 노  
드 B에 전송하는 과정과, 상기 노드 B가 DPCH를 통해 상기 DSCH 및 DCH의 TFCI를  
UE에 전송하는 과정으로 이루어진다.

**【대표도】**

도 10

**【색인어】**

TFCI, DSCH, DCH, hard split, logical split

**【명세서】****【발명의 명칭】**

분할모드에 따른 전송형식 조합표시 비트의 전송 장치 및 방법 {apparatus and method for transmitting TFCI bits in cdma communication system}

**【도면의 간단한 설명】**

도 1. Hard split 방법에 해당하는 (16,5)coding을 나타내는 도면

도 2. Hard split 방법을 위한 기지국간의 신호메시지 및 데이터 전송을 나타내는 도면

도 3. Logical split 방법을 위한 기지국간의 신호메시지 및 데이터 전송을 나타내는 도면

도 4. DSCH용 TFCI와 DCH용 TFCI정보 비트를 수신하여 각각 서로 다른 부호화 방법으로 부호화 하여 전송하는 본 발명의 도면

도 5. 본 발명의 도4의 부호기를 나타내는 도면

도 6. 본 발명의 부호화 방법으로 부호화된 부호화 심볼을 수신하여 복호화 하는 수신기를 나타내는 도면

도 7. 본 발명의 수신기에 사용되는 복호기를 나타내는 도면

도 8. 다운링크 DCH의 신호 전송 포맷을 나타내는 도면

도 9. 서로 다른 방법으로 코딩된 부호화 심볼을 멀티플레싱 하는 방법을 나타내는 도면

도10. SRNC 와 DRNC가 동일하지 않는 경우 상기 Logical split 방법을 위한  
기지국간의 신호메시지 및 데이터 전송을 나타내는 본 발명의 도면

도11. 본 발명의 DRNC의 동작과정을 나타내는 도면

도12. SRNC의 동작과정을 나타내는 도면

도13. 도 8. DRNC로부터 SRNC로 전송되는 정보를 담고 있는 Control Frame  
의 구조의 한 예를 보여 주는 도면

도 14. 본 발명에 따른 복호기의 구조를 나타내는 도면

도 15. 본 발명에 따른 가변적인 길이를 가지는 역하다마드 변환기를 도시  
하는 도면

도 16. 길이 8인 월시부호에 따른 일반적인 역하다마드 연산과정을 나타내  
는 그림

도 17. 도 15에서 각각의 단계에 대한 장치를 나타내는 도면

### 【발명의 상세한 설명】

#### 【발명의 목적】

#### 【발명이 속하는 기술분야 및 그 분야의 종래기술】

<18> 본 발명은 비동기 이동통신 방식에서 사용하는 채널중에 하향 공유 채널의  
데이터 전송시 사용되는 전송형식 조합표시 비트를 전송하는 장치 및 방법에 관  
한 것이다.

- <19> 통상적으로 하향 공유채널(Downlink Shared Channel, 이하 'DSCH'라 칭한다)은 여러 사용자가 시간을 분할하여 사용하는 공유채널이다. 상기 DSCH는 각 사용자마다 전용 채널(Dedicated Channel: 이하 'DCH'라 칭한다.)과 함께 설정될 수 있다. 상기 DCH는 DPCCH와 DPDCH를 포함하고 있다. 특히 DSCH를 위한 물리 제어채널로 전용물리채널(Dedicated Physical Control Channel: 이하 'DPCCH'라 칭한다)가 사용된다. 따라서 DPCCH는 해당 DCH와 DSCH를 위한 물리 제어 채널로 사용되며 제어 신호중의 하나인 전송형식 조합표시 비트(Transport Format Combination Indicator: 이하 'TFCI'라 칭한다)[TFCI]에 대한 정보가 DCH와 DSCH를 위하여 동시에 필요하다.
- <20> 상기 TFCI는 물리채널을 통해 전송되는 데이터 정보량을 나타내는 정보이다.
- <21> 상기 TFCI는 10bit로 구성된 정보를 30bit로 coding을 이용하여 나타내어진다. 즉 데이터 정보량에 관한 정보는 10bit로 표현되고 이를 물리 채널로 전송시에는 30bit로 구성하여 전송하게 된다.
- <22> DPCCH에 DCH를 위한 TFCI와 DSCH를 위한 TFCI를 동시에 전송하는 방법은 크게 두가지 방법으로 구분된다. 첫째가 하드 스플리트(이하 Hard split라 칭한다) 방법이고 둘째가 논리 스플리트(이하 Logical split라 칭한다) 방법이다.
- <23> 상기 DCH를 위한 TFCI는 TFCI(field 1) 또는 제1TFCI라 칭하고 DSCH를 위한 TFCI는 TFCI(field 2) 또는 제2TFCI라 칭한다.

- <24> 상기 Logical split 방법의 경우 상기 TFCI(field 1)과 TFCI(field 2)를 하나의 TFCI로 상기 (30,10)천공된 리드뮬러 코드(sub-code sencond order Reed Muller Code)로 부호화(coding)하여 전송한다. 상기 TFCI(field 1)과 TFCI(field 2)는 정보 10bit를 1:9 또는 2:8 또는 3:7 또는 4:6 또는 5:5 또는 6:4 또는 7:3 또는 8:2 또는 9:1 등으로 나누어 정보를 나타낸 후, 10bit를 하나의 블럭코딩(block coding), 즉 (30,10)천공된 리드뮬러 코드(sub-code sencond order Reed Muller Code)로 부호화하여 전송한다.
- <25> 상기 Hard split 방법의 경우 상기 TFCI(field 1)과 TFCI(field 2)는 각각 5 비트씩으로 나타내어지고, 각각의 정보는 (15,5)천공된 이중직교부호(Bi-Orthogonal code)를 이용하여 출력한 후, 상기 각각의 15 비트를 멀티플렉싱하여 30 비트로 만들어 물리채널로 전송한다.
- <26> 도 1은 상기 Hard split 방법에 따른 송신기의 구조를 나타내는 도면이다.
- <27> 먼저 5비트의 DCH를 위한 TFCI(field 1)비트들이 (15,5)Bi-Orthogonal 부호기 100에 입력되면 상기 부호기 100은 상기 5비트의 DCH를 위한 TFCI(field 1)비트들을 부호화하여 15심볼의 부호화 심볼들을 멀티플렉서 110으로 출력하고, 동시에 5비트의 DSCH를 위한 TFCI(field 2)비트들이 (15,5)Bi-Orthogonal 부호기 105에 입력되면 상기 부호기 105는 상기 5비트의 DCH를 위한 TFCI(field 1)비트들을 부호화하여 15심볼의 부호화 심볼들을 멀티플렉서 110으로 출력한다. 그러면, 상기 멀티플렉서 110은 상기 부호기 100에서 출력된 15심볼의 부호화 심볼들과 상기 부호기 105에서 출력된 15심볼의 부호화 심볼들을 시간적으로 멀티플렉싱하여 30심볼을 배열하여 출력한다. 그러면, 상기 출력된 30심볼은 다시 멀티플

렉서 120으로 입력되어, 동시에 입력된 기타신호들과 시간적으로 멀티플렉싱되어 전후 확산기 130으로 출력되면, 상기 확산기 130은 확산부호 생성기 135로부터 입력되어진 확산부호로 확산하여 출력하고, 상기와 같은 확산되어진 신호들은 다시 스크램블러 140에 입력되고, 상기 스크램블러 140은 스크램블링 부호 생성기 145로부터 입력되어진 스크램블링 부호로 스크램블되어 출력되어진다.

<28>        도 2은 기존의 3GPP(3RD Generation PartnerShip Project)에서 정의한 상기 Hard split 방법을 위한 기지국간의 신호메시지 및 데이터 전송을 나타내는 도면이다.

<29>        상기 도 2의 단계 101에서 Serving RNC(이하 'SRNC'라 칭한다)의 Radio Link Control(RLC)는 SRNC의 Mac-d에 전송할 data를 전송한다. 이 때 전송되는 primitive는 Mac-d data-req이다.

<30>        상기 도 2의 단계 102에서 SRNC의 Mac-d는 CRNC의 Mac-c에 RLC로부터 수신한 data를 전송한다. 이 때 전송되는 primitive는 Mac-c/sh-data-req이다.

<31>        상기 도 2의 단계 103에서 CRNC는 단계 2에서 SRNC의 Mac-d에서 수신한 data를 위한 전송시간을 결정한 후 (scheduling) data와 해당 TFI를 함께 Node B의 L1으로 전송한다. 이 때 전송되는 primitive는 Mphy-data-req이다.

<32>        상기 도 2의 단계 104에서 SRNC의 Mac-d에서 Node B의 L1으로 전송될 DCH의 data와 해당 TFI를 전송한다. 이 때 전송되는 primitive는 Mphy-data-req이다.

<33>        상기 도 2의 단계 103과 단계 104에서 전송되는 data들은 서로 독립적으로 전송되며 Node-B 의 L1은 DCH와 DSCH용으로 분할 되어 있는 TFCI를 생성한다.

- <34> 상기 도 2의 단계 103과 단계 104에서 data와 TFI는 data frame protocol을 이용하여 전송된다.
- <35> 상기 도 2의 단계 105에서 Node B의 L1은 UE의 L1으로 DSCH data를 Physical DSCH(이하 'PDSCH'라 칭한다)로 전송한다.
- <36> 상기 도 2의 단계 106에서 Node B의 L1은 UE의 L1으로 DPCH를 이용하여 TFCI를 전송한다. 이때 상기 단계 3과 단계 4에서 수신한 TFI를 이용하여 생성한 TFCI를 각각 DCH용 DSCH용 field를 이용하여 전송한다.
- <37> 도 3은 상기 Logical split 방법을 위한 기지국간의 신호메시지 및 데이터 전송을 나타내는 도면이다.
- <38> 상기 도 3의 단계 201에서 RNC의 RLC는 RNC의 Mac-d에 전송할 DSCH data를 전송한다. 이 때 전송되는 primitive는 Mac-d data-req이다.
- <39> 상기 도 3의 단계 202에서 RNC의 Mac-d는 RNC의 Mac-c에 RLC로부터 수신한 data를 전송한다. 이 때 전송되는 primitive는 Mac-c/sh-data-req이다.
- <40> 상기 도 3의 단계 203에서는 단계 202에서 RNC의 Mac-d에서 수신한 data를 위한 전송시간을 결정한 후 (scheduling) 해당 TFCI를 RNC의 Mac-d로 전송한다.
- <41> 상기 도 3의 단계 204에서 RNC의 Mac-c는 Node B의 L1로 DSCH 데이터를 전송한다. 이 때 전송되는 DSCH 데이터는 단계 203에서 미리 결정된 (scheduled) 시간에 전송된다.
- <42> 상기 도 3의 단계 205에서 RNC의 Mac-d에서 Node B의 L1으로 DSCH를 위한 TFCI를 결정하여 전송한다. 이 때 전송되는 primitive는 Mphy-data-req이다.

- <43> 상기 도 3의 단계 206에서 RNC의 Mac-d에서 Node B의 L1으로 전송될 DCH의 data와 DCH를 위한 TFCI를 결정하여 전송한다. 이 때 전송되는 프리미티브(primitive)는 Mphy-data-req이다.
- <44> 상기 도 3의 단계 204에서 전송되는 DSCH data와 단계 205에서 전송되는 TFCI는 상기 단계 203에서 결정된 시간과 관계를 갖는다. 즉 단계 205에서 전송되는 TFCI는 단계 204에서 전송되는 DSCH 데이터가 PDSCH로 전송되기 직전 Frame에 DPCCH로 UE에게 전송되어 진다.
- <45> 상기 도 3의 단계 204와 단계 205와 단계 206에서 전송되는 데이터 및 TFCI는 프레임 프로토콜(Frame protocol)을 이용하여 전송된다. 특히 단계 206에서 전송되는 TFCI는 제어프레임(Control frame)을 통하여 전송된다.
- <46> 상기 도 3의 단계 207에서 Node B의 L1은 UE의 L1으로 DSCH 데이터를 PDSCH로 전송한다.
- <47> 상기 도 3의 단계 208에서 Node B의 L1은 UE의 L1으로 DPCH를 이용하여 TFCI를 전송한다. 이때 상기 단계 205과 단계 206에서 수신한 각각의 TFCI 또는 TFI를 이용하여 하나의 TFCI를 생성하여 DPCCH를 이용하여 전송한다.
- <48> 상기 설명에 의하면 Logical split 방법의 경우 Mac-c는 Mac-d에 DSCH scheduling 정보 및 해당 DSCH의 TFCI 정보를 전송한다. 이것은 DSCH와 DCH에 관한 TFCI를 하나의 코딩 방법으로 부호화하기 때문에 Mac-d에서 이 정보를 동시에 노드 B(Node B)의 L1으로 보내야 하기 때문이다. 따라서 Mac-d에 전송할 데이터

가 있는 경우 데이터를 Mac-c로 전송한 후 Mac-c로 부터의 스케줄링(Scheduling) 정보 및 TFCI 정보를 수신할 때까지 지연이 생길 수 있다. 또한 Mac-c와 Mac-d가 Iur상에 분리되어 있는 경우 즉 Mac-c는 DRNC에 Mac-d는 SRNC에 있는 경우 Iur 상으로 스케줄링 정보 및 TFCI 정보를 주고 받기 때문에 더 큰 지연이 생길 수 있다.

<49> 상기 설명에 의하면 hard split 방법의 경우 Logical split 방법의 경우에 비해 Mac-c에서의 스케줄링 이후에 Mac-d로의 정보전송이 필요하지 않아 지연(delay)을 줄일 수 있다. 이것은 hard split의 경우 Node B에서 DCH용 TFCI와 DSCH용 TFCI를 각각 독립적으로 코딩할 수 있기 때문에 가능하다. 또한 Mac-c와 Mac-d가 Iur 상에 분리되어 있는 경우 즉 Mac-c는 DRNC에 Mac-d는 SRNC에 있는 경우 Iur상으로 scheduling 정보를 주고 받지 않기 때문에 장점이 있다. 그러나 상기 설명에 의하면 DCH용과 DSCH용 TFCI는 정보량은 각각 5bit씩 나누어 지게 되어 각각 최대 32개의 정보를 나타낼 수 있고 따라서 DSCH를 위한 TFCI정보가 32개 이상인 경우는 hard split를 사용할 수 없는 단점이 있다.

#### 【발명이 이루고자 하는 기술적 과제】

<50> 따라서 본 발명의 목적은 Logical split 방법의 경우 Mac-c와 Mac-d가 Iur 상에 분리되어 있는 경우 (즉, DRNC와 SRNC가 분리되어 있는 경우) Scheduling 정보를 Mac-c에서 Mac-d로 전송하는 장치 및 방법을 제공함에 있다.

- <51> 본 발명의 다른 목적은 TFCI 송신 방법에 있어서 가변적인 길이를 가지는 월시 부호기에 대한 역하다마드 변환기를 사용하여, TFCI 수신 방법 및 장치를 최적화하는 방법 및 장치를 제공함에 있다.
- <52> 본 발명의 다른 목적은 DSCH의 두 개의 TFCI 송신 방법인 hard split와 logical split을 구분하여 사용하도록 하는 신호 메시지를 전송에 관한 방법을 제공함에 있다.

### 【발명의 구성 및 작용】

- <53> 이하 본 발명의 일 실시 예를 첨부된 도면을 참조하여 상세히 설명하면 다음과 같다.
- <54> 한편, 본 발명의 실시 예는 본 발명의 주된 내용을 구체화하기 위하여 필요 한 것이며, 본 발명의 내용을 제한하지는 않는다. 또한, 본 발명의 실시 예를 설명함에 있어 앞에서 설명한 구성요소와 동일한 동작을 하는 다른 도면의 구성요 소는 이전과 동일한 도면 참조번호를 사용하고 있음에 주의하여야 할 것이다.
- <55> 본 발명은 hard split 방법의 경우 DSCH를 위한 정보비트와 DCH를 위한 정보비트의 수가 총 10bit 중에서 1:9 또는 2:8 또는 3:7 또는 4:6 또는 5:5 또는 6:4 또는 7:3 또는 8:2 또는 9:1 등으로 나누어 정보를 나타낸 후 각각에 대하여 코딩(coding)을 적용할 수 있도록 하는 장치 및 방법을 제공함에 있다.
- <56> 먼저 물리계층에서 한 프레임에는 30개의 TFCI 부호 심볼이 전송되는데 이 는 부호율이 1/3이다. 상기 TFCI 정보비트가 상기와 같이 일정비로 나누어질 때,

상기의 일정비와 같은 비율로 부호심볼의 개수를 나누어 각각의 부호율을 유지하는 것이 바람직하다. 예를들어, 10비트의 입력비트들이 1:9로 나누어질 경우, 30심볼의 출력심볼은 3:27로 나누어지고, 10비트의 입력비트들이 2:8로 나누어질 경우, 30심볼의 출력심볼은 6:24로 나누어지고, 10비트의 입력비트들이 3:7로 나누어질 경우, 30심볼의 출력심볼은 9:21로 나누어지고, 10비트의 입력비트들이 4:6로 나누어질 경우, 30심볼의 출력심볼은 12:18로 나누어지는 것이 바람직하다. 따라서,

<57> 정보량의 비율이 1:9일 경우, 1비트를 입력받아 3개의 부호심볼을 출력하는 (3,1)부호와 9비트를 입력받아 27개의 부호심볼을 출력하는 (27,9)부호가 필요하고, 정보량의 비율이 2:8일 경우, 2비트를 입력받아 6개의 부호심볼을 출력하는 (6,2)부호와 8비트를 입력받아 24개의 부호심볼을 출력하는 (24,8)부호가 필요하고, 정보량의 비율이 3:7일 경우, 3비트를 입력받아 9개의 부호심볼을 출력하는 (9,3)부호와 7비트를 입력받아 21개의 부호심볼을 출력하는 (21,7)부호가 필요하고, 정보량의 비율이 4:6일 경우, 4비트를 입력받아 12개의 부호심볼을 출력하는 (12,4)부호와 6비트를 입력받아 18개의 부호심볼을 출력하는 (18,6)부호가 필요하다. 따라서, 상기 10가지의 부호기들이 성능면에서 우수하고, 하드웨어의 복잡도를 위해서 상기의 10가지 부호기들이 한가지 구조로 동작할 수 있는 부호기가 필요하다.

<58> 통상적으로 선형 오류정정부호(Linear Error Correcting Code)의 성능을 타내는 척도(measure)로서는 오류정정부호의 부호어(codeword)의 해밍 거리(Hamming distance) 분포가 있는데, 이는 각각의 부호어에서 0이 아닌 심볼의 개

수를 의미한다. 즉, 0111이 어떤 부호어라면 이 부호어에 포함된 1의 개수, 즉, 해밍거리는 3이다. 이 때, 여러 부호어의 해밍거리 값들 중 가장 작은 값을 최소 거리( $d_{min}$ ; minimum distance)라고 칭한다. 상기 선형 오류정정부호(Linear Error Correcting Code)에 있어서 상기의 최소거리가 클수록 오류정정 성능이 우수한데, 이는 참조문헌 'The Theory of Error-Correcting Codes' - F.J.Macwilliams, N.J.A. Sloane, North-Holland에서 상세히 개시하고 있다.

<59> 또한, 하드웨어 복잡도를 위해서 상기 서로 다른 길이의 부호기를 한가지 부호기 구조로 동작시키기 위해서는 가장 큰 길이의 부호, 즉, (32,10)부호를 Shortening하는 것이 바람직하다. 그러나, 상기 Shortening방법을 사용하기 위해서는 부호심볼의 천공이 필요한데, 상기 부호를 천공함에 있어, 천공의 위치에 따라서 상기 부호의 최소거리가 달라진다. 따라서, 상기 천공된 부호가 최적의 최소거리를 가지도록 하는 천공위치를 구하는 것이 바람직하다. 예를 들어, 상기 여러개의 부호 중, (6,2)부호로써 가장 최적의 부호는 (3,2)심플렉스 부호를 2번 반복하여 사용하는 것이 최소거리관점에서 가장 바람직하다. 이 때, 상기 (3,2)심플렉스의 입력정보비트와 출력되는 (3,2)심플렉스 부호어 간의 관계는 하기 <표1>과 같다.

<60> 【표 1】

입력정보비트	(3,2)심플렉스 부호어
00	000
01	101
10	011
11	110

<61> 따라서, 상기 (3,2)심플렉스 부호어를 2회 반복하게 되면, 입력정보비트와 출력되는 (3,2)심플렉스 부호어 간의 관계는 하기 <표2>과 같다.

<62> 【표 2】

입력정보비트	(3,2)심플렉스의 2회 반복 부호어
00	000 000
01	101 101
10	011 011
11	110 110

<63> 그러나 상기와 같은 (3,2)심플렉스 부호어를 2회 반복한 부호어는 기존에 사용하고 있는 (16,4)Reed Muller부호를 Shortening함으로써 구현할수 있다.

Shortening방법에 대해서 예를 들어 설명하면, 먼저, (16,4)Reed Muller부호는 입력정보비트수인 4개의 길이 16인 기저 부호어의 선형결합인데, 이 중 2비트만 입력이 된다고 함은 4개의 길이 16인 기저 부호어중 2개의 기저 부호어만의 선형결합을 사용하고, 나머지는 사용하지 않는다는 것이다. 또한, 상기와 같이 기저 부호어의 사용을 제한한 후, 상기 16인 길이중 10개의 심볼들을 천공하면, 상기 (16,4)부호기를 사용하여, (6,2)부호기를 동작시킬 수 있다. 하기에 나타나는 <표3>는 상기에서 설명한 Shortening을 설명한다.

<64>

【표 3】

입력정보비트	부호어														
0000	0(*)	0	0	0	0(*)	0	0	0	0(*)	0(*)	0(*)	0(*)	0(*)	0(*)	0(*)
0001	0(*)	1	0	1	0(*)	1	0	1	0(*)	1(*)	0(*)	1(*)	0(*)	1(*)	0(*)
0010	0(*)	0	1	1	0(*)	0	1	1	0(*)	0(*)	1(*)	1(*)	0(*)	0(*)	1(*)
0011	0(*)	1	1	0	0(*)	1	1	0	0(*)	1(*)	1(*)	0(*)	0(*)	1(*)	1(*)
0100	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1
0101	0	1	0	1	1	0	1	0	0	1	0	1	1	0	1
0110	0	0	1	1	1	1	0	0	0	1	1	1	1	0	0
0111	0	1	1	0	1	0	0	1	0	1	1	0	1	0	1
1000	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
1001	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1
1010	0	0	1	1	0	0	1	1	1	0	0	1	1	0	0
1011	0	1	1	0	0	1	1	0	1	0	0	1	1	0	1
1100	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0
1101	0	1	0	1	1	0	1	0	1	0	1	0	0	1	0
1110	0	0	1	1	1	0	0	1	1	0	0	0	0	1	1
1111	0	1	1	0	1	0	0	1	1	0	0	1	0	1	0

<65> 상기 <표 3>을 참조하면, 먼저 모든  $(16, 4)$ 부호어는 짚은 체로 쓰여있는 4개의 길이 16인 기저부호어의 선형결합이다. 이 때, 상기  $(6, 2)$ 부호를 얻기 위해서는, 상기 4개의 기저 부호어 중 상위 2개의 부호어만을 사용하면, 나머지 하위 12개의 부호어는 자동적으로 쓰이지 않게 되고, 상위 4개의 부호어만을 사용하게 된다. 또한 상위 4개의 부호어 중 길이를 6인 부호어를 만들려면 10개의 심볼을 천공하여야 하는데, 상기 <표 3>에서 (\*)로 표시된 부분을 천공하고 나머지 6개의 부호심볼들을 모으면 상기 <표 2>에서 나타낸  $(3, 2)$ 심플렉스의 2회 반복 부호어를 얻을 수 있다. 따라서, 하기에서는  $(32, 10)$ Sub-code of the Second order Reed Muller code를 Shortening하여, 정보량의 비가 1:9일 경우에 사용되어지는  $(3, 1)$ 최적부호와  $(27, 9)$ 최적부호를 만드는 부호기의 구조와, 정보량의 비가 2:8일 경우에 사용되어지는  $(6, 2)$ 최적부호와  $(24, 8)$ 최적부호를 만드는 부호기의 구조와, 정보량의 비가 3:7일 경우에 사용되어지는  $(9, 3)$ 최적부호와  $(21, 7)$ 최적부

호를 만드는 부호기의 구조와, 정보량의 비가 4:6일 경우에 사용되어지는 (12,4) 최적부호와 (18,6)최적부호를 만드는 부호기의 구조와, 정보량의 비가 5:5일 경우에 사용되어지는 (15,5)최적부호와 (15,5)최적부호를 만드는 부호기의 구조에 대해서 실시예를 통해서 설명되어질 것이다.

<66> 하기에 나타날 실시예는 hard split 방법에서 hard split 방법의 경우 각각 5bit로 구성되는 정보량을 Logical의 경우와 같이 10bit를 1:9 또는 2:8 또는 3:7 또는 4:6 또는 5:5 또는 6:4 또는 7:3 또는 8:2 또는 9:1 등으로 나누어 정보를 나타낸 후 각각에 대하여 coding을 적용할 수 있도록 하는 장치 및 방법을 나타낸다.

<67> 실 시 예

<68> 도 4은 상기 실시예에 따른 송신기의 구조를 도시한다. 도 4를 참조하면, 상기 정보량의 비에 따른 배분되어진 DSCH용 TFCI비트와 DCH용 TFCI비트가 부호기 400과 405에 각각 입력된다. 여기서 상기 DSCH용 TFCI비트는 TFCI(field 1) 또는 제1TFCI가 될 수 있으며, DCH용 TFCI비트는 TFCI(field2) 또는 제2TFCI비트가 된다. 상기 DSCH용 TFCI비트는 제1TFCI비트 발생기450에서 발생되며, DCH용 TFCI비트는 제2TFCI비트 발생기 455에 발생된다. 그리고 상기 제1 및 제2TFCI비트는 정보량의 비에 따라 상기한 바와 같이 각각 다른 비율을 갖는 TFCI비트들을 생성한다. 또한 상기 정보량의 비에 따른 부호어의 길이 설정값인 부호길이 정보를 나타내는 제어신호가 부호기 400과 부호기 405에 입력된다. 상기 부호길이

정보는 부호길이정보 발생기460에서 발생되며, 이때 상기 부호길이 정보는 상기 제1TFCI 및 제2TFCI비트의 길이에 따라 가변되는 값을 갖는다.

<69> 이 때, 정보량의 비가 6:4일 경우, 상기 6비트의 DSCH용 TFCI비트가 부호기 400에 입력됨과 동시에 부호기 400은 6비트의 입력을 받아 18심볼의 부호어를 출력하는 (18,6)부호기로 동작하도록 하는 제어신호를 입력받아서 (18,6)부호기로 동작한 후 18심볼의 부호화 심볼들을 출력하고, 상기 4비트의 DCH용 TFCI비트가 부호기 400에 입력됨과 동시에 부호기 400은 4비트의 입력을 받아 12심볼의 부호어를 출력하는 (12,4)부호기로 동작하도록 하는 제어신호를 입력받아서 (12,4)부호기로 동작한 후 12심볼의 부호화 심볼들을 출력하게 된다. 또한, 정보량의 비가 7:3일 경우, 상기 7비트의 DSCH용 TFCI비트가 부호기 400에 입력됨과 동시에 부호기 400은 7비트의 입력을 받아 21심볼의 부호어를 출력하는 (21,7)부호기로 동작하도록 하는 제어신호를 입력받아서 (21,7)부호기로 동작한 후 21심볼의 부호화 심볼들을 출력하고, 상기 3비트의 DCH용 TFCI비트가 부호기 400에 입력됨과 동시에 부호기 400은 3비트의 입력을 받아 9심볼의 부호어를 출력하는 (9,3)부호기로 동작하도록 하는 제어신호를 입력받아서 (9,3)부호기로 동작한 후 9심볼의 부호화 심볼들을 출력하게 된다. 또한, 정보량의 비가 8:2일 경우, 상기 8비트의 DSCH용 TFCI비트가 부호기 400에 입력됨과 동시에 부호기 400은 8비트의 입력을 받아 24심볼의 부호어를 출력하는 (24,8)부호기로 동작하도록 하는 제어신호를 입력받아서 (24,8)부호기로 동작한 후 18심볼의 부호화 심볼들을 출력하고, 상기 2비트의 DCH용 TFCI비트가 부호기 400에 입력됨과 동시에 부호기 400은 2비트의 입력을 받아 6심볼의 부호어를 출력하는 (6,2)부호기로 동작하도록 하는 제

어신호를 입력받아서 (6,2)부호기로 동작한 후 6심볼의 부호화 심볼들을 출력하게 된다. 또한, 정보량의 비가 9:1일 경우, 상기 9비트의 DSCH용 TFCI비트가 부호기 400에 입력됨과 동시에 부호기 400은 9비트의 입력을 받아 27심볼의 부호어를 출력하는 (27,9)부호기로 동작하도록하는 제어신호를 입력받아서 (27,9)부호기로 동작한 후 27심볼의 부호화 심볼들을 출력하고, 상기 1비트의 DCH용 TFCI비트가 부호기 400에 입력됨과 동시에 부호기 400은 1비트의 입력을 받아 3심볼의 부호어를 출력하는 (3,1)부호기로 동작하도록하는 제어신호를 입력받아서 (3,1)부호기로 동작한 후 3심볼의 부호화 심볼들을 출력하게 된다.

- <70>      도 5는 상기 부호기 400과 부호기 405의 구조를 나타낸다. 도 5를 참조하여 각각의 정보량의 비에 따라서 설명한다.
- <71>      먼저 정보량의 비가 1:9 의 경우를 경우를 살펴본다.
- <72>      정보량의 비가 1:9인 경우에 부호기 400의 경우 (3,1)부호기로 동작하고, 부호기 405는 (27,9)부호기로 동작하게 된다. 따라서, 부호기 400의 동작과 부호기 405의 동작을 각각 살펴보면 다음과 같다.
- <73>      먼저 상기 부호기 400의 동작을 살펴본다.
- <74>      1비트의 입력비트가 부호기 400에 입력되면, 상기 입력비트는 a0로 하고, 나머지, a1,a2,a3,a4,a5,a6,a7,a8,a9은 0로 채운다. 그러면 상기, 입력비트 a0는 승산기 510으로, 입력비트 a1는 승산기 512으로, 입력비트 a2는 승산기 514으로, 입력비트 a3는 승산기 516으로, 입력비트 a4는 승산기 518으로, 입력비트 a5는 승산기 520으로, 입력비트 a6는 승산기 522으로, 입력비트 a7는 승산기 524으로,

입력비트 a8는 승산기 526으로, 입력비트 a9는 승산기 528으로 입력된다. 그러면, 이와 동시에 월시부호 생성기 500은 기저부호어  $W_1 = 1010101010101011010101010101010100$ 를 생성하여 승산기 510로 출력하면, 상기 승산기 510은 심볼단위로 상기 부호어와 입력비트 a0를 승산하여 배타적가산기 540으로 출력한다. 그리고, 상기 월시부호 생성기 500는 그 외의 기저부호어  $W_2, W_4, W_8, W_{16}$ 을 생성하여 각각 승산기 512, 514, 516, 518로 출력하고,  $a_{11} 1$  부호생성기 502는 전부 1인 기저 부호어를 생성하여 승산기 520로 출력하고, 마스크 생성기 504는 그 외의 기저부호어  $M_1, M_2, M_4, M_8$ 을 생성하여 각각 승산기 522, 524, 526, 528로 출력하지만, 상기 승산기 512, 514, 516, 518, 520, 522, 524, 526, 528에 입력되어진 상기 입력비트  $a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9$ 이 0이기 때문에, 상기 승산기 512, 514, 516, 518, 520, 522, 524, 526, 528는 0를 출력하기 때문에 상기 승산기들이 배타적 가산기 540에 출력하여도 영향을 주지 못한다. 즉, 상기 배타적 가산기 540이 승산기 510, 512, 514, 516, 518, 520, 522, 524, 526, 528으로 부터의 출력값들을 모두 배타적 가산하여 출력하여도, 상기 승산기 510으로 부터의 출력값이 그대로 출력된다. 그러면 상기 배타적 가산기 540으로부터 출력된 32개의 심볼은 천공기 560으로 입력된다. 그러면 이와 동시에 부호길이 정보가 제어기 550에 입력되면 제어기 550은 부호길이에 해당하는 천공위치에 대한 제어신호를 천공기 560에 출력하고, 그러면, 상기 천공기 560은 상기 제어기 500에 성 출력된 제어신호에 따라 상기 입력된 0번째부터 31번째 까지의 총 32개의 부호심볼중, 1, 3, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,

18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31번째 부호심볼을 천공

하여 32부호심볼중 29심볼이 천공된 3개의 부호화 심볼을 출력한다.

<75> 두 번째로 부호기 405의 동작을 살펴본다.

<76> 9비트의 입력비트가 부호기 400에 입력되면, 상기 입력비트들은 각각 a0, a2, a3, a4, a5, a6, a7, a8로 하고, 나머지, a9은 0로 채운다. 그러면 상기, 입력비트 a0는 승산기 510으로, 입력비트 a1는 승산기 512으로, 입력비트 a2는 승산기 514으로, 입력비트 a3는 승산기 516으로, 입력비트 a4는 승산기 518으로, 입력비트 a5는 승산기 520으로, 입력비트 a6는 승산기 522으로, 입력비트 a7는 승산기 524으로, 입력비트 a8는 승산기 526으로, 입력비트 a9는 승산기 528으로 입력된다. 그러면, 이와 동시에 월시부호 생성기 500은 기저부호어W1 = 101010101010110101010101010100를 생성하여 승산기 510로 출력하고, 기저부호어W2 = 01100110011001101100110011001100를 생성하여 승산기 512로 출력하고, 기저부호어W4 = 000111100001111000111100001111100를 생성하여 승산기 514로 출력하고, 기저부호어W8 = 00000001111111000000011111111111100를 생성하여 승산기 516로 출력하고, 기저부호어W16 = 000000000000000111111111111101를 생성하여 승산기 518로 출력하면, 상기 승산기 510은 심볼단위로 상기 기저부호어 W1과 입력비트 a0를 승산하여 배타적가산기 540으로 출력하고, 상기 승산기 512은 심볼단위로 상기 기저부호어 W2과 입력비트 a1를 승산하여 배타적가산기 540으로 출력하고, 상기 승산기 514은 심볼단위로 상기 기저부호어 W4과 입력비트 a2를 승산하여 배타적가산기 540으로 출력하고, 상기 승산기 516은 심볼단위로 상기 기저부호어 W8과 입력비트 a3를 승산하여 배타적가산기 540으로 출력하고, 상기

승산기 518은 심볼단위로 상기 기저부호어 W16과 입력비트 a4를 승산하여 배타적  
가산기 540으로 출력한다. 또한, a11 1 부호생성기 502는 전부 1인 길이 32인 기  
저 부호어를 생성하여 승산기 520로 출력하면, 상기 승산기 520은 심볼단위로 상  
기 기저부호어 a11 1 부호어와 입력비트 a5를 승산하여 배타적가산기 540으로 출  
력한다. 또한, 마스크 생성기 540은 기저부호어  $M1 = 0101\ 0000\ 1100\ 0111\ 1100$   
 $0001\ 1101\ 1101$ 를 생성하여 승산기 522로 출력하고, 기저부호어  $M2 = 0000\ 0011$   
 $1001\ 1011\ 1011\ 0111\ 0001\ 1100$ 를 생성하여 승산기 524로 출력하고, 기저부호어  
 $M4 = 0001\ 0101\ 1111\ 0010\ 0110\ 1100\ 1010\ 1100$ 를 생성하여 승산기 526로 출력하  
면, 상기 승산기 522은 심볼단위로 상기 기저부호어 M1과 입력비트 a6를 승산하  
여 배타적가산기 540으로 출력하고, 상기 승산기 524은 심볼단위로 상기 기저부  
호어 M2과 입력비트 a7를 승산하여 배타적가산기 540으로 출력하고, 상기 승산기  
526은 심볼단위로 상기 기저부호어 M4과 입력비트 a8를 승산하여 배타적가산기  
540으로 출력한다. 그리고, 상기 마스크 생성기 504는 그 외의 기저부호어 M8을  
생성하여 각각 승산기 528로 출력하지만, 상기 승산기 528에 입력되어진 상기  
입력비트 a9이 0이기 때문에, 상기 승산기 528는 0를 출력하기 때문에 상기 승산  
기가 배타적 가산기 540에 출력하여도 영향을 주지 못한다. 즉, 상기 배타적 가  
산기 540이 승산기 528으로부터의 출력값들을 모두 배타적 가산하여 출력하여도,  
상기 승산기 510, 512, 514, 516, 518, 520, 522, 524, 526으로부터의 출력값을 배타적  
가산한 결과만이 출력된다. 그러면, 상기 배타적 가산기 540으로부터 출력된 32  
개의 심볼은 천공기 560으로 입력된다. 그러면 이와 동시에 부호길이정보가 제어  
기 550에 입력되면 제어기 550은 부호길이에 해당하는 천공위치에 대한 제어신호

를 천공기 560에 출력하고, 그러면, 상기 천공기 560은 상기 제어기 500에 성 출력된 제어신호에 따라상기 입력된 0번째부터 31번째 까지의 총 32개의 부호심볼 중, 0, 2, 8, 19, 20번째 부호심볼을 천공하여 32부호심볼중 5심볼이 천공된 27개의 부호화 심볼을 출력한다.

<77> 두 번째로 정보량의 비가 2:8인 경우의 동작을 살펴본다.

<78> 정보량의 비가 2:8인 경우에 부호기 400의 경우 (6,2)부호기로 동작하고, 부호기 405는 (24,8)부호기로 동작하게 된다. 따라서, 부호기 400의 동작과 부호기 405의 동작을 각각 살펴보면 다음과 같다.

<79> 먼저 부호기 400의 동작을 살펴본다.

<80> 2비트의 입력비트가 부호기 400에 입력되면, 상기 입력비트는 a0,a1으로 하고, 나머지, a2, a3, a4, a5, a6, a7, a8, a9은 0로 채운다. 그러면 상기, 입력비트 a0는 승산기 510으로, 입력비트 a1는 승산기 512으로, 입력비트 a2는 승산기 514으로, 입력비트 a3는 승산기 516으로, 입력비트 a4는 승산기 518으로, 입력비트 a5는 승산기 520으로, 입력비트 a6는 승산기 522으로, 입력비트 a7는 승산기 524으로, 입력비트 a8는 승산기 526으로, 입력비트 a9는 승산기 528으로 입력된다. 그러면, 이와 동시에 월시부호 생성기 500은 기저부호어  $W_1 = 101010101010101101010101010100$ 를 생성하여 승산기 510로 출력하면, 상기 승산기 510은 심볼단위로 상기 부호어와 입력비트 a0를 승산하여 배타적가산기 540으로 출력하고, 기저부호어  $W_2 = 01100110011001101100110011001100$ 를 생성하여 승산기 512로 출력하면, 상기 승산기 512은 심볼단위로 상기 부호어와 입력비트 a1를 승산하여 배타적가산기 540으로 출력한다. 그리고, 상기 월시부호 생성기

500는 그외의 기저부호어 W4, W8, W16을 생성하여 각각 승산기 514, 516, 518로 출력하고, all 1 부호생성기 502는 전부 1인 기저 부호어를 생성하여 승산기 520로 출력하고, 마스크 생성기 504는 그 외의 기저부호어 M1, M2, M4, M8을 생성하여 각각 승산기 522, 524, 526, 528로 출력하지만, 상기 승산기 514, 516, 518, 520, 522, 524, 526, 528에 입력되어진 상기 입력비트 a2,a3,a4,a5,a6,a7,a8,a9이 0이기 때문에, 상기 승산기 514, 516, 518, 520, 522, 524, 526, 528는 0를 출력하기 때문에 상기 승산기들이 배타적 가산기 540에 출력하여도 영향을 주지 못한다. 즉, 상기 배타적 가산기 540이 승산기 510, 512, 514, 516, 518, 520, 522, 524, 526, 528으로 부터의 출력값들을 모두 배타적 가산하여 출력하여도, 상기 승산기 510과 승산기 512으로 부터의 출력값의 배타적가산값만이 출력된다. 그러면 상기 배타적 가산기 540으로부터 출력된 32개의 심볼은 천공기 560으로 입력된다. 그러면 이와 동시에 부호길이정보가 제어기 550에 입력되면 제어기 550은 부호길이에 해당하는 천공위치에 대한 제어신호를 천공기 560에 출력하고, 그러면, 상기 천공기 560은 상기 제어기 500에 성 출력된 제어신호에 따라상기 입력된 0번째부터 31번째 까지의 총 32개의 부호심볼중, 3, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 번째 부호심볼을 천공하여 32부호심볼중 26심볼이 천공된 6개의 부호화 심볼을 출력한다.

<81> 다음으로 부호기 405의 동작을 살펴본다.

<82> 8비트의 입력비트가 부호기 400에 입력되면, 상기 입력비트들은 각각 a0, a2, a3, a4, a5, a6, a7로 하고, 나머지, a8, a9은 0로 채운다. 그러면 상기, 입

력비트 a0는 승산기 510으로, 입력비트 a1는 승산기 512으로, 입력비트 a2는 승산기 514으로, 입력비트 a3는 승산기 516으로, 입력비트 a4는 승산기 518으로, 입력비트 a5는 승산기 520으로, 입력비트 a6는 승산기 522으로, 입력비트 a7는 승산기 524으로, 입력비트 a8는 승산기 526으로, 입력비트 a9는 승산기 528으로 입력된다. 그러면, 이와 동시에 월시부호 생성기 500은 기저부호어  $W_1 = 101010101010101101010101010100$ 를 생성하여 승산기 510로 출력하고, 기저부호어  $W_2 = 01100110011001101100110011001100$ 를 생성하여 승산기 512로 출력하고, 기저부호어  $W_4 = 0001110000111100011110000111100$ 를 생성하여 승산기 514로 출력하고, 기저부호어  $W_8 = 00000001111111000000011111111111100$ 를 생성하여 승산기 516로 출력하고, 기저부호어  $W_{16} = 000000000000000111111111111101$ 를 생성하여 승산기 518로 출력하면, 상기 승산기 510은 심볼단위로 상기 기저부호어  $W_1$ 과 입력비트 a0를 승산하여 배타적가산기 540으로 출력하고, 상기 승산기 512은 심볼단위로 상기 기저부호어  $W_2$ 과 입력비트 a1를 승산하여 배타적가산기 540으로 출력하고, 상기 승산기 514은 심볼단위로 상기 기저부호어  $W_4$ 과 입력비트 a2를 승산하여 배타적가산기 540으로 출력하고, 상기 승산기 516은 심볼단위로 상기 기저부호어  $W_8$ 과 입력비트 a3를 승산하여 배타적가산기 540으로 출력하고, 상기 승산기 518은 심볼단위로 상기 기저부호어  $W_{16}$ 과 입력비트 a4를 승산하여 배타적가산기 540으로 출력한다. 또한, a11 1 부호생성기 502는 전부 1인 길이 32인 기저 부호어를 생성하여 승산기 520로 출력하면, 상기 승산기 520은 심볼단위로 상기 기저부호어 a11 1 부호어와 입력비트 a5를 승산하여 배타적가산기 540으로 출력한다. 또한, 마스크 생

성기 540은 기저부호어M1 = 0101 0000 1100 0111 1100 0001 1101 1101를 생성하여 승산기 522로 출력하고, 기저부호어M2 = 0000 0011 1001 1011 1011 0111 0001 1100를 생성하여 승산기 524로 출력하면, 상기 승산기 522은 십볼단위로 상기 기저부호어 M1과 입력비트 a6를 승산하여 배타적가산기 540으로 출력하고, 상기 승산기 524은 십볼단위로 상기 기저부호어 M2과 입력비트 a7를 승산하여 배타적가산기 540으로 출력한다. 그리고, 상기 마스크 생성기 504는 그 외의 기저부호어 M4, M8을 생성하여 각각 승산기 528로 출력하지만, 상기 승산기 526, 528에 각각 입력되어진 상기 입력비트 a8, a9이 0이기 때문에, 상기 승산기 526, 528는 0을 출력하기 때문에 상기 승산기가 배타적 가산기 540에 출력하여도 영향을 주지 못한다. 즉, 상기 배타적 가산기 540이 승산기 526, 528으로부터의 출력값들을 모두 배타적 가산하여 출력하여도, 상기 승산기 510, 512, 514, 516, 518, 520, 522, 524으로부터의 출력값을 배타적가산한 결과만이 출력된다. 그러면, 상기 배타적 가산기 540으로부터 출력된 32개의 십볼은 천공기 560으로 입력된다. 그러면 이와 동시에 부호길이정보가 제어기 550에 입력되면 제어기 550은 부호길이에 해당하는 천공위치에 대한 제어신호를 천공기 560에 출력하고, 그러면, 상기 천공기 560은 상기 제어기 500에 성 출력된 제어신호에 따라상기 입력된 0번째부터 31번째 까지의 총 32개의 부호심볼중, 1, 7, 13, 15, 20, 25, 30, 31번째 부호심볼을 천공하여 32부호심볼중 8심볼이 천공된 24개의 부호화 십볼을 출력한다.

<83> 세 번째로 정보량의 비가 3:7 의 경우의 동작을 살펴본다.

<84> 정보량의 비가 3:7인 경우에 부호기 400의 경우 (9,3)부호기로 동작하고, 부호기 405는 (21,7)부호기로 동작하게 된다. 따라서, 부호기 400의 동작과 부호기 405의 동작을 각각 살펴보면 다음과 같다.

<85> 먼저 부호기 400의 동작을 살펴본다.

<86> 3비트의 입력비트가 부호기 400에 입력되면, 상기 입력비트는  $a_0, a_1, a_2$ 으로 하고, 나머지,  $a_3, a_4, a_5, a_6, a_7, a_8, a_9$ 은 0로 채운다. 그러면 상기, 입력비트  $a_0$ 는 승산기 510으로, 입력비트  $a_1$ 는 승산기 512으로, 입력비트  $a_2$ 는 승산기 514으로, 입력비트  $a_3$ 는 승산기 516으로, 입력비트  $a_4$ 는 승산기 518으로, 입력비트  $a_5$ 는 승산기 520으로, 입력비트  $a_6$ 는 승산기 522으로, 입력비트  $a_7$ 는 승산기 524으로, 입력비트  $a_8$ 는 승산기 526으로, 입력비트  $a_9$ 는 승산기 528으로 입력된다. 그러면, 이와 동시에 월시부호 생성기 500은 기저부호어  $W_1 = 1010101010101101010101010100$ 를 생성하여 승산기 510로 출력하면, 상기 승산기 510은 십불단위로 상기 부호어와 입력비트  $a_0$ 를 승산하여 배타적가산기 540으로 출력하고, 기저부호어  $W_2 = 01100110011001101100110011001100$ 를 생성하여 승산기 512로 출력하면, 상기 승산기 512은 십불단위로 상기 부호어와 입력비트  $a_1$ 를 승산하여 배타적가산기 540으로 출력하고, 기저부호어  $W_4 = 0001110000111100011110000111100$ 를 생성하여 승산기 514로 출력하면, 상기 승산기 514은 십불단위로 상기 부호어와 입력비트  $a_2$ 를 승산하여 배타적가산기 540으로 출력한다. 그리고, 상기 월시부호 생성기 500은 그외의 기저부호어  $W_8, W_{16}$ 을 생성하여 각각 승산기 516, 518로 출력하고, all 1 부호생성기 502는 전부 1인 기저 부호어를 생성하여 승산기 520로 출력하고, 마스크 생성기 504는 그

외의 기저부호어 M1, M2, M4, M8을 생성하여 각각 승산기 522, 524, 526, 528로 출력하지만, 상기 승산기 516, 518, 520, 522, 524, 526, 528에 입력되어진 상기 입력비트 a<sub>2</sub>, a<sub>3</sub>, a<sub>4</sub>, a<sub>5</sub>, a<sub>6</sub>, a<sub>7</sub>, a<sub>8</sub>, a<sub>9</sub>이 0이기 때문에, 상기 승산기들이 배타적 가산기 540에 출력하여도 영향을 주지 못한다. 즉, 상기 배타적 가산기 540이 승산기 510, 512, 514, 516, 518, 520, 522, 524, 526, 528으로 부터의 출력값들을 모두 배타적 가산하여 출력하여도, 상기 승산기 510, 승산기 512과 승산기 514으로 부터의 출력값의 배타적 가산값만이 출력된다. 그러면 상기 배타적 가산기 540으로부터 출력된 32개의 심볼은 천공기 560으로 입력된다. 그러면 이와 동시에 부호길이정보가 제어기 550에 입력되면 제어기 550은 부호길이에 해당하는 천공위치에 대한 제어신호를 천공기 560에 출력하고, 그러면, 상기 천공기 560은 상기 제어기 500에 상기 출력된 제어신호에 따라 상기 입력된 0번째부터 31번째 까지의 총 32개의 부호심볼중, 7, 8, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31번째 부호심볼을 천공하여 32부호심볼중 23심볼이 천공된 9개의 부호화 심볼을 출력한다.

<87> 다음으로 부호기 405의 동작을 살펴본다.

<88> 7비트의 입력비트가 부호기 400에 입력되면, 상기 입력비트들은 각각 a<sub>0</sub>, a<sub>2</sub>, a<sub>3</sub>, a<sub>4</sub>, a<sub>5</sub>, a<sub>6</sub>로 하고, 나머지, a<sub>7</sub>, a<sub>8</sub>, a<sub>9</sub>은 0로 채운다. 그러면 상기, 입력비트 a<sub>0</sub>는 승산기 510으로, 입력비트 a<sub>1</sub>는 승산기 512으로, 입력비트 a<sub>2</sub>는 승산기 514으로, 입력비트 a<sub>3</sub>는 승산기 516으로, 입력비트 a<sub>4</sub>는 승산기 518으로, 입력비트 a<sub>5</sub>는 승산기 520으로, 입력비트 a<sub>6</sub>는 승산기 522으로, 입력비트 a<sub>7</sub>는 승산기 524으

로, 입력비트 a8는 승산기 526으로, 입력비트 a9는 승산기 528으로 입력된다. 그러면, 이와 동시에 월시부호 생성기 500은 기저부호어  $W_1 = 1010101010101101010101010100$ 를 생성하여 승산기 510로 출력하고, 기저부호어  $W_2 = 01100110011001101100110011001100$ 를 생성하여 승산기 512로 출력하고, 기저부호어  $W_4 = 00011110000111100011110000111100$ 를 생성하여 승산기 514로 출력하고, 기저부호어  $W_8 = 0000000111111100000000111111111111101$ 를 생성하여 승산기 516로 출력하고, 기저부호어  $W_{16} = 00000000000000001111111111111101$ 를 생성하여 승산기 518로 출력하면, 상기 승산기 510은 심볼단위로 상기 기저부호어  $W_1$ 과 입력비트 a0를 승산하여 배타적가산기 540으로 출력하고, 상기 승산기 512은 심볼단위로 상기 기저부호어  $W_2$ 과 입력비트 a1를 승산하여 배타적가산기 540으로 출력하고, 상기 승산기 514은 심볼단위로 상기 기저부호어  $W_4$ 과 입력비트 a2를 승산하여 배타적가산기 540으로 출력하고, 상기 승산기 516은 심볼단위로 상기 기저부호어  $W_8$ 과 입력비트 a3를 승산하여 배타적가산기 540으로 출력하고, 상기 승산기 518은 심볼단위로 상기 기저부호어  $W_{16}$ 과 입력비트 a4를 승산하여 배타적가산기 540으로 출력한다. 또한, a11 1 부호생성기 502는 전부 1인 길이 32인 기저 부호어를 생성하여 승산기 520로 출력하면, 상기 승산기 520은 심볼단위로 상기 기저부호어 a11 1 부호어와 입력비트 a5를 승산하여 배타적가산기 540으로 출력한다. 또한, 마스크 생성기 540은 기저부호어  $M_1 = 0101 0000 1100 0111 1100 0001 1101 1101$ 를 생성하여 승산기 522로 출력하면, 상기 승산기 522은 심볼단위로 상기 기저부호어  $M_1$ 과 입력비트 a6를 승산하여 배타적가산기 540으로 출력한다. 그리고, 상기 마스크 생성기

504는 그 외의 기저부호어 M2, M4, M8을 생성하여 각각 승산기 524, 526, 528로 출력하지만, 상기 승산기 524, 526, 528에 각각 입력되어진 상기 입력비트 a7, a8, a9이 0이기 때문에, 상기 승산기 524, 526, 528는 0를 출력하기 때문에 상기 승산기가 배타적 가산기 540에 출력하여도 영향을 주지 못한다. 즉, 상기 배타적 가산기 540이 승산기 510, 512, 514, 516, 518, 520, 522, 524, 526, 528으로 부터의 출력값들을 모두 배타적 가산하여 출력하여도, 상기 승산기 510, 512, 514, 516, 518, 520, 522으로 부터의 출력값을 배타적 가산한 결과만이 출력된다. 그러면, 상기 배타적 가산기 540으로부터 출력된 32개의 심볼은 천공기 560으로 입력된다. 그러면 이와 동시에 부호길이정보가 제어기 550에 입력되면 제어기 550은 부호길이에 해당하는 천공위치에 대한 제어신호를 천공기 560에 출력하고, 그러면, 상기 천공기 560은 상기 제어기 500에 성 출력된 제어신호에 따라상기 입력된 0번째부터 31번째 까지의 총 32개의 부호심볼중, 0, 1, 2, 3, 4, 5, 7, 12, 18, 21, 24번째 부호심볼을 천공하여 32부호심볼중 11심볼이 천공된 21개의 부호화 심볼을 출력한다.

<89> 네 번째로 정보량의 비가 4:6 의 경우의 동작을 살펴본다.

<90> 정보량의 비가 4:6인 경우에 부호기 400의 경우 (12,3)부호기로 동작하고, 부호기 405는 (18,6)부호기로 동작하게 된다. 따라서, 부호기 400의 동작과 부호기 405의 동작을 각각 살펴보면 다음과 같다.

<91> 먼저 부호기 400의 동작을 살펴본다.

<92> 4비트의 입력비트가 부호기 400에 입력되면, 상기 입력비트는 a0,a1,a2,a3으로 하고, 나머지 a4, a5, a6, a7, a8, a9은 0로 채운다. 그러면 상기, 입력비

트 a0는 승산기 510으로, 입력비트 a1는 승산기 512으로, 입력비트 a2는 승산기 514으로, 입력비트 a3는 승산기 516으로, 입력비트 a4는 승산기 518으로, 입력비트 a5는 승산기 520으로, 입력비트 a6는 승산기 522으로, 입력비트 a7는 승산기 524으로, 입력비트 a8는 승산기 526으로, 입력비트 a9는 승산기 528으로 입력된다. 그러면, 이와 동시에 월시부호 생성기 500은 기저부호어  $W_1 = 10101010101010110101010101010100$ 를 생성하여 승산기 510로 출력하면, 상기 승산기 510은 십불단위로 상기 부호어와 입력비트 a0를 승산하여 배타적가산기 540으로 출력하고, 기저부호어  $W_2 = 01100110011001101100110011001100$ 를 생성하여 승산기 512로 출력하면, 상기 승산기 512은 십불단위로 상기 부호어와 입력비트 a1를 승산하여 배타적가산기 540으로 출력하고, 기저부호어  $W_4 = 00011110000111100011110000111100$ 를 생성하여 승산기 514로 출력하면, 상기 승산기 514은 십불단위로 상기 부호어와 입력비트 a2를 승산하여 배타적가산기 540으로 출력하고, 기저부호어  $W_8 = 000000011111111000000000 1111111100$ 를 생성하여 승산기 516로 출력하면, 상기 승산기 514은 십불단위로 상기 부호어와 입력비트 a3를 승산하여 배타적가산기 540으로 출력한다. 그리고, 상기 월시부호 생성기 500은 그외의 기저부호어  $W_{16}$ 을 생성하여 각각 승산기 518로 출력하고, a11 1부호생성기 502는 전부 1인 기저 부호어를 생성하여 승산기 520로 출력하고, 마스크 생성기 504는 그 외의 기저부호어 M1, M2, M4, M8을 생성하여 각각 승산기 522, 524, 526, 528로 출력하지만, 상기 승산기 518, 520, 522, 524, 526, 528에 입력되어진 상기 입력비트 a4,a5,a6,a7,a8,a9이 0이기 때문에, 상기 승산기들이 배타적 가산기 540

에 출력하여도 영향을 주지 못한다. 즉, 상기 배타적 가산기 540이 승산기 510, 512, 514, 516, 518, 520, 522, 524, 526, 528으로 부터의 출력값들을 모두 배타적 가산하여 출력하여도, 상기 승산기510, 승산기512, 승산기514과 승산기516으로 부터의 출력값의 배타적가산값만이 출력된다. 그러면 상기 배타적 가산기 540으로부터 출력된 32개의 심볼은 천공기 560으로 입력된다. 그러면 이와 동시에 부호길이정보가 제어기 550에 입력되면 제어기 550은 부호길이에 해당하는 천공 위치에 대한 제어신호를 천공기 560에 출력하고, 그러면, 상기 천공기 560은 상기 제어기 500에 상기 출력된 제어신호에 따라 상기 입력된 0번째부터 31번째 까지의 총 32개의 부호심볼중, 0, 1, 2, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31번째 부호심볼을 천공하여 32부호심볼중 20심볼이 천공된 12개의 부호화 심볼을 출력한다.

<93> 다음으로 부호기 405의 동작을 살펴본다.

<94> 6비트의 입력비트가 부호기 400에 입력되면, 상기 입력비트들은 각각 a0, a1, a2, a3, a4, a5로 하고, 나머지a6, a7, a8, a9은 0로 채운다. 그러면 상기, 입력비트 a0는 승산기 510으로, 입력비트 a1는 승산기 512으로, 입력비트 a2는 승산기 514으로, 입력비트 a3는 승산기 516으로, 입력비트 a4는 승산기 518으로, 입력비트 a5는 승산기 520으로, 입력비트 a6는 승산기 522으로, 입력비트 a7는 승산기 524으로, 입력비트 a8는 승산기 526으로, 입력비트 a9는 승산기 528으로 입력된다. 그러면, 이와 동시에 월시부호 생성기 500은 기저부호어W1 = 10101010101010110101010101010100를 생성하여 승산기 510로 출력하고,

기저부호어W2 = 01100110011001101100110011001100를 생성하여 승산기 512로 출력하고, 기저부호어W4 = 00011110000111100011110000111100를 생성하여 승산기 514로 출력하고, 기저부호어W8 = 00000001111111000000001111111100를 생성하여 승산기 516로 출력하고, 기저부호어W16 = 000000000000000011111111111101를 생성하여 승산기 518로 출력하면, 상기 승산기 510은 심볼단위로 상기 기저부호어 W1과 입력비트 a0를 승산하여 배타적가산기 540으로 출력하고, 상기 승산기 512은 심볼단위로 상기 기저부호어 W2과 입력비트 a1를 승산하여 배타적가산기 540으로 출력하고, 상기 승산기 514은 심볼단위로 상기 기저부호어 W4과 입력비트 a2를 승산하여 배타적가산기 540으로 출력하고, 상기 승산기 516은 심볼단위로 상기 기저부호어 W8과 입력비트 a3를 승산하여 배타적가산기 540으로 출력하고, 상기 승산기 518은 심볼단위로 상기 기저부호어 W16과 입력비트 a4를 승산하여 배타적가산기 540으로 출력한다. 또한, a11 1 부호생성기 502는 전부 1인 길이 32인 기저 부호어를 생성하여 승산기 520로 출력하면, 상기 승산기 520은 심볼단위로 상기 기저부호어 a11 1 부호어와 입력비트 a5를 승산하여 배타적가산기 540으로 출력한다. 그리고, 상기 마스크 생성기 504는 기저부호어 M1, M2, M4, M8을 생성하여 각각 승산기 522, 524, 526, 528로 출력하지만, 상기 승산기 522, 524, 526, 528에 각각 입력되어진 상기 입력비트 a6, a7, a8, a9이 0이기 때문에, 상기 승산기 522, 524, 526, 528는 0를 출력하기 때문에 상기 승산기가 배타적 가산기 540에 출력하여도 영향을 주지 못한다. 즉, 상기 배타적 가산기 540이 승산기 510, 512, 514, 516, 518, 520, 522, 524, 526, 528으로부터의 출력값들을 모두 배타적 가산하여 출력하여도, 상기 승산

기 510, 512, 514, 516, 518, 520, 522으로 부터의 출력값을 배타적 가산한 결과 만이 출력된다. 그러면, 상기 배타적 가산기 540으로부터 출력된 32개의 심볼은 천공기 560으로 입력된다. 그러면 이와 동시에 부호길이 정보가 제어기 550에 입력되면 제어기 550은 부호길이에 해당하는 천공위치에 대한 제어신호를 천공기 560에 출력하고, 그러면, 상기 천공기 560은 상기 제어기 500에 상기 출력된 제어신호에 따라 상기 입력된 0번째부터 31번째 까지의 총 32개의 부호심볼 중, 0, 7, 9, 11, 16, 19, 24, 25, 26, 27, 28, 29, 30, 31번째 부호심볼을 천공하여 32부호심볼 중 14심볼이 천공된 18개의 부호화 심볼을 출력한다.

<95> 상기한 바와 같이 정보량의 비가 각각 9:1, 8:2, 7:3, 또는 6:4인 경우에는 위의 설명에 의하여 각각 부호기 400과 405의 동작이 이해될 수 있을 것이다.

<96> 상기와 같은 동작 후 부호기 400과 405로부터 출력되어지는 부호화 심볼들은 각각 멀티플렉서 410에 인가되어 시간적으로 멀티플렉싱되어지면, 30심볼의 멀티플렉싱된 신호들이 출력되어진다.

<97> 다음은 DSCH 와 DCH를 위한 코딩을 각각 실시한 후 멀티플렉서 410에서 멀티플렉싱 하는 방법에 대하여 설명한다. 상기 멀티플렉서 410은 상기 부호기 400과 405에서 출력되는 부호화 심볼을 가능한 균일하게 분포되도록 멀티플렉싱하여 30 비트를 배열한다.

<98> DCH와 DSCH를 위한 TFCI 비트를 각각  $m$  비트와  $n$  비트를 사용한 것으로 정한다.

<99> 이때  $m$ 과  $n$ 이 가질 수 있는 값은  $(m,n)=1:9$  또는  $2:8$  또는  $3:7$  또는  $4:6$  또는  $5:5$  또는  $6:4$  또는  $7:3$  또는  $8:2$  또는  $9:1$ 이다.

<100> 우선  $m$ 값이  $n$ 값보다 큰 경우를 고려한다.

<101>  $n$ 값이  $m$ 값보다 큰 경우에도  $n$ 과  $m$  값을 교환하여 하기 방법을 이용하면 DCH와 DSCH의 배열을 얻을 수 있다.

<102> 상기 서술된 코딩 방법에 의하면 DCH와 DSCH를 위한 TFCI 비트가 각각  $m$ 비트와  $n$ 비트인 경우 코딩 후에 생성된 비트 수는 각각  $m*3$ 비트와  $n*3$ 비트이다.

<103> 따라서 생성된 부호화 심볼을 전송하기 위한 위치를 선정하기 위하여 우선 DPCCH로 전송될 30비트를 10비트씩 나눈 후 각각의 10비트에 DCH를 위한  $m*3$ 비트를 3등분한  $m$ 비트와 DSCH를 위한  $n*3$ 비트를 3등분한  $n$ 비트를 배열한다.

<104> 다음은 주어진 10비트에 DCH를 위한  $m$ 비트와 DSCH를 위한  $n$ 비트를 배열하는 방법에 관한 실시예이다.

<105>  $L$ 을 10비트 중  $L$  번째 비트를 나타내도록 하자.

<106> 다음을 정의하자.

$$<107> \text{【수학식 1】 } F(k) = \lfloor \frac{m}{n} * k \rfloor \quad k=0,1,2,\dots, n$$

$$<108> \text{【수학식 2】 } G(k) = \lfloor \frac{F(k)-F(k-1)}{2} \rfloor \quad k=0,1,2,3,\dots, n$$

<109> 상기 <수학식 1> 및 <수학식 2>에서  $\lfloor X \rfloor$  는  $x$  보다 작거나 같은 정수 중 가장 큰 값을 나타내고  $\lfloor X \rfloor$  는  $x$  보다 크거나 같은 정수 중 가장 작은 값을 나타낸다.

<110> 상기 수식 2에서  $F(-1)$ 은 0으로 정의한다. 즉

<111>  $F(-1)=0$

<112> 상기 수식을 이용하여 DCH를 위한  $m$  비트와 DSCH를 위한  $n$  비트를 배열하는 방법을 서술 하면 다음의 <수학식 3>과 같다. 10개의  $L$  값들 중  $n$  개의 다음의  $L$  값들에는 DSCH를 위한 비트를 순서대로 배열 한다.

<113> 【수학식 3】  $L = F(1-1) + G(1) + 1$

<114> 상기 수식에서 1 값은 다음의 범위를 갖는다.  $1 = 1 \leq n$

<115> 10개의  $L$  값들 중 상기 식 3에 주어진 값들 이외의  $L$  값들에는 DCH를 위한  $m$  개의 비트를 나열하면 된다. 즉 하기 <수학식 4>와 같이 표시할 수 있다.

<116> 【수학식 4】  $F(1-2) + G(1-1) + 1 = L \leq F(1-1) + G(1) + 1-1$

<117> 각각의 경우 즉  $m:n$  이 9:1, 8:2, 7:3, 6:4, 5:5 인 경우  $F(k)$  및  $G(k)$ 를 나타내면 다음의 <표 4>와 같다.

<118>

【표 4】

m:n	F(k)	F(1)	F(2)	F(3)	F(4)	F(5)
	G(k)	G(1)	G(2)	G(3)	G(4)	G(5)
	DSCH의 위치					
5:5	1	2	3	4	5	
	1	1	1	1	1	
	2	4	6	8	10	
6:4	1	3	4	6		
	1	1	1	1		
	2	4	7	9		
7:3	2	4	7			
	1	1	1			
	2	5	8			
8:2	4	8				
	2	2				
	3	8				
9:1	9					
	4					
	5					

<119> 도 9는  $m:n=6:4$ 인 경우에 DCH를 위한 TFCI 비트와 DSCH를 위한 TFCI 비트를 DPCCH 30비트에 대응시키는 관계를 설명하는 도면이다. 상기 <표 4>에서와 같이  $m:n=6:4$ 인 경우 DSCH의 위치는 L 값이 2, 4, 7, 9인 경우에 위치한다.

<120> 그러면, 상기 멀티플렉싱되어진 신호들은 다시 멀티플렉서 420에 인가되어 전력제어비트(TPC), 파일럿비트와 같은 기타신호와 시간적으로 멀티플렉싱되져 출력되어진다. 상기와 같은 멀티플렉싱은 도 8에 나타나 있는 바와 같이 시간적 혼합이 일어난다. 그러면, 상기 출력되어진 멀티플렉싱된 심볼들은 확산기 430으로 입력되면 아와 동시에, 확산부호가 확산부호 생성기 435로부터 확산부호가 입력되어지고, 채널구분을 위해 심볼단위로 확산부호로 채널확산되어져 칩단위로 출력되어진다. 그러면, 상기 채널확산되어진 신호들은 다시 스크램블러440에 입

력이 되고, 이와 동시에 스크램블링 부호발생기 445로부터 스크램블링 부호가 입력되어져 상기 입력된 신호를 상기 입력되어진 스크램블링 부호로 스크램블링하여 출력되어진다.

- <121>      도 6은 상기 실시예에 따른 수신기의 구조를 도시한다.
- <122>      도 6을 참조하여 설명하면, 먼저, 수신신호가 디스크램블러 640에 입력된다. 그러면, 이와 동시에 스크램블링 부호발생기 645로부터 스크램블링 부호가 입력되어져 상기 입력된 수신신호를 상기 입력되어진 스크램블링 부호로 디스크램블링하여 출력한다. 그러면, 상기 출력되어진 디스크램블링되어진 심볼들은 역확산기 630으로 입력되면 이와 동시에, 확산부호가 확산부호 생성기 635로부터 확산부호가 입력되어지고, 역확산되어져 심볼단위로 출력되어진다.
- <123>      그러면, 상기 역확산되어진 수신신호는 디멀티플렉서 620에 입력되어져 디멀티플렉싱되어져, 전력제어비트(TPC), 파일럿비트, 피드백 신호와 같은 기타신호와 TFCI를 분리하여 출력되면 상기 분리되어진 TFCI 부호화 심볼들은 다시 디멀티플렉서 610에 입력되어진다. 이와 동시에 상기 DSCH용 TFCI비트와 DCH용 TFCI비트의 정보량비율에 따른 부호길이에 대한 제어정보가 디멀티플렉서 610에 입력되어져 상기 비율에 따라 DSCH용 TFCI부호심볼과 DCH용 TFCI부호심볼로 분리되어져 각각의 복호기 600과 복호기 605로 입력되어진다. 그러면, 동시에 상기 DSCH용 TFCI비트와 DCH용 TFCI비트의 정보량비율에 따른 부호길이에 대한 제어정보가 각각의 복호기 600과 복호기 605로 입력되어져 각각에 해당하는 부호에 따른 복호과정을 거친후 각각 DSCH용 TFCI비트와 DCH용 TFCI비트를 출력한다.
- <124>      도 7은 상기 도 6의 복호기 600과 복호기 605의 구조를 나타낸다.

<125> 도 7을 참조하여 설명하면, 상기 수신 심볼이 복호기에 입력되어지면, 상기 수신심볼은 0삽입기 700에 입력되어지고, 이와 동시에 부호 길이정보가 제어기 730에 입력되어진다. 상기 제어기 730는 상기 수신 심볼의 부호길이에 따른 천공 위치를 저장하고 있다가 이에 대한 제어정보를 0삽입기 700에 출력하는데, 부호율이 (3,1)이면, 29개의 천공위치에 대한 정보를, 부호율이 (6,2)이면, 26개의 천공위치에 대한 정보를, 부호율이 (9,3)이면, 23개의 천공위치에 대한 정보를, 부호율이 (12,4)이면, 20개의 천공위치에 대한 정보를, 부호율이 (18,6)이면, 14개의 천공위치에 대한 정보를, 부호율이 (21,7)이면, 11개의 천공위치에 대한 정보를, 부호율이 (24,8)이면, 8개의 천공위치에 대한 정보를, 부호율이 (27,9)이면, 5개의 천공위치에 대한 정보를 출력한다. 각각의 경우에 대하여 천공위치는 상기 부호기 설명에서 제시한 바와 같다. 그러면 상기, 0삽입기 700은 상기 천공 위치에 대한 제어정보에 따라서 천공위치에 0를 삽입하여 길이 32인 심볼열을 출력한다. 그러면, 상기 심볼열은 역하다마드 변환기 720과 승산기 702, 704, 706에 각각 입력되어진다. 상기 승산기 720, 704, 706으로 입력되어진 신호들은 마스크 생성기 710으로부터 생성되어진 모든 경우의 마스크 함수들이 출력되어져 각각 승산되어진 후 출력되어지는데, 상기 출력되어진 심볼들은 스위치 752, 754, 756으로 각각 입력되어진다. 그러면 이와 동시에, 상기 제어기 730는 상기 입력되어진 부호길이 정보에 따른 마스크함수의 사용여부에 대한 제어정보를 스위치 752, 754, 756으로 각각 출력하는데, (3,1), (6,2), (9,3), (12,4), (18,6) 부호기는 마스크 함수를 사용하지 않기 때문에, 상기 제어정보에 따라 스위치 752, 754, 756은 전부 연결을 끊게 된다. 그리고, (21,7)부호기의 경우는 기저



1020010010150

출력 일자: 2002/2/28

마스크 함수를 1개만 쓰기 때문에, 스위치 752만 연결이 되고, 상기와 같이 부호율에 따라 사용되어지는 마스크 함수 개수에 따라서 스위치가 제어를 한다. 그러면, 상기 역하다마드 변환기 720, 724, 726은 상기 각각 입력된 32개의 심볼을 역하다마드 변환하여, 상기 입력신호에 승산되어진 마스크 함수의 색인을 나타내는 0 (상기 역하다마드 변환기 720으로 입력되어진 신호는 어떤 마스크 함수도 승산되어지지 않았기 때문에)과 길이 월시부호와의 상관도 중 가장 높은 상관도를 가지는 월시부호의 색인과 상기 상관도를 계산하여 상관도 비교기 740에 입력한다. 그러면 상기 740은 상기 입력된 상관도들을 비교하여 가장 높은 상관도를 가지는 것에 대한 마스크 색인과 월시부호 색인을 결합하여 출력한다.

<126> 한편, 전술한 바와 같은 여러 가지 길이의 부호에 의한 부호화에 대응하여 각각의 정보량에 대한 복호를 모두 수행할 수 있는 복호기의 구조와 동작에 대해서도 제안되어야 할 것이다.

<127> 이하 본 발명에서 제안하고자 하는 복호화 방법을 상세히 설명하면 다음과 같다.

<128> 본 발명의 실시 예에 따른 복호기의 동작을 살펴보면, (6,2)부호기에 대응한 복호기로 동작할 때에는 길이 4인 월시부호기에 대한 역하다마드 변환기가 사용되고, (9,3)부호기에 대응한 복호기로 동작할 때에는 길이 8인 월시부호기에 대한 역하다마드 변환기가 사용된다. (12,4)부호기에 대응한 복호기로 동작할 때에는 길이 16인 월시부호기에 대한 역하다마드 변환기가 사용되고, (15,5)부호기로 대응한 복호기로 동작할 때에는 길이 16인 월시부호기에 대한 역하다마드 변환기가 사용된다. 그 외에 (18,6), (21,7), (24,8), (27,9) 및 (30,10)부호기에

대응한 복호기로 동작할 때에는 길이 32인 월시부호기에 대한 역하다마드 변환기가 사용된다. 상기와 같은 동작을 하기 위해서는 가변길이에 대해 동작 가능한 역하다마드 변환기 구조를 가져야한다. 본 발명에서는 상기와 같은 가변길이에 대해 동작 가능한 역하다마드 변환기 구조를 제공하고, 하기의 실시 예에서 나타내기로 한다.

<129> 본 발명의 실시 예에 따른 복호화 방법을 상세히 설명하기에 앞서 본 발명에 적용되는 통상적인 역하다마드 변환기의 연산 구조를 도 16을 참조하여 설명하면 다음과 같다.

<130> 도 16은 길이 8인 월시부호기에 따른 일반적인 역하다마드 변환기의 연산 구조를 도시한다. 일반적으로 길이  $2^n$ 인 월시부호기에 대한 역하다마드 변환기는 n개의 단계로 구성되어져 있다. 각각의 단계에서는 하나의 열에 대응하여 입력되어지는 2개의 입력신호를 가산하거나 감산하는 과정을 진행한다.

<131> 상기 도 16을 참조하여 길이 8인 월시부호기에 따른 역하다마드의 연산 구조를 설명한다.

<132> 단계 1에서는  $r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8$ 을 입력으로 하여 두 신호간의 가산과 감산을 수행한다. 즉, 첫 번째 열에서는  $r_1$ 과  $r_2$ 에 대해서, 두 신호간의 가산과 감산을 하여  $r_1+r_2, r_1-r_2$ 를 출력하여 길이 2인 월시부호에 대한 상관값을 계산한다. 그 외에도,  $r_3$ 과  $r_4$ 에 대해서, 두 신호간의 가산과 감산을 하여  $r_3+r_4, r_3-r_4$ 를 출력하고,  $r_5$ 과  $r_6$ 에 대해서, 두 신호간의 가산과 감산을 하여  $r_5+r_6, r_5-r_6$ 을 출력하며,  $r_7$ 과  $r_8$ 에 대해서, 두 신호간의 가산과 감산을 하여  $r_7+r_8, r_7-r_8$ 을 출력한다. 그러면, 단계 1을 통해 출력되어진 8개의 연산되어진

신호들에 대해서 단계2를 진행[하면,]한다. 상기 단계 2에서는  $r_1+r_2$ 와  $r_3+r_4$ 에 대해서, 두 신호를 가산하여  $(r_1 + r_2) + (r_3 + r_4)$ 를 출력하고,  $r_1-r_2$ 와  $r_3-r_4$ 에 대해서, 두 신호를 가산하여  $(r_1 - r_2) + (r_3 - r_4)$ 를 출력[하고,]한다. 또한,  $r_1+r_2$ 와  $r_3+r_4$ 에 대해서, 두 신호를 감산하여  $(r_1 + r_2) - (r_3 + r_4)$ 를 출력하고,  $r_1-r_2$ 와  $r_3-r_4$ 에 대해서, 두 신호를 감산하여  $(r_1 - r_2) - (r_3 - r_4)$ 를 출력하여 길이 2인 월시부호에 대한 상관값을 계산[하고,]한다. 그 외에도,  $r_5+r_6$ 와  $r_7+r_8$ 에 대해서, 두 신호를 가산하여  $(r_5 + r_6) + (r_7 + r_8)$ 를 출력하고,  $r_5-r_6$ 와  $r_7-r_8$ 에 대해서, 두 신호를 가산하여  $(r_5 - r_6) + (r_7 - r_8)$ 를 출력하고,  $r_5+r_6$ 와  $r_7+r_8$ 에 대해서, 두 신호를 감산하여  $(r_5 + r_6) - (r_7 + r_8)$ 을 출력하고,  $r_5-r_6$ 와  $r_7-r_8$ 에 대해서, 두 신호를 감산하여  $(r_5 - r_6) - (r_7 - r_8)$ 을 출력한다. 상기와 유사한 과정으로 연산을 하여 단계 3을 진행하게 되면, 길이 8인 월시부호와의 모든 상관값을 출력할 수 있다. 상기와 같은 예를 통해보면, 일반적인 길이  $2^n$ 인 월시 부호기에 따른 역하다마드 변환기의 단계 i의 과정은 단계 i에 입력되어진  $2^n$ 의 신호를  $2^i$ 개씩 순서대로 한 블록으로 묶어서  $2^{n-i}$ 개의 블록으로 나누면, 각각의 블록내에서 k번째 신호와  $k+2^i-1$ 번째 신호를 차례대로 모두 가산하고, 그 이후에 다시 k번째 신호와  $k+2^i-1$ 번째 신호를 차례대로 모두 감산하는 과정으로 이루어진다. 상기와 같은 각각의 단계에 대한 연산과정을 수행하여 역하다마드 변환과정을 수행하면, 단계 i번째 과정이 끝난 직후, 출력되어지는 신호들 중 처음  $2^i$ 개의 출력신호를 보면 역하다마드 변환기에 입력되어진 입력신호의 처음  $2^i$ 개의 출력신호를 길이  $2^i$ 인 월시부호와

의 모든 상관값이 순서대로 출력되어짐을 알 수 있다. 하기의 제2실시 예에서는 상기와 같은 연산성질에 따른 복호기에 대한 실시 예를 나타낸다.

<133>      제2 실시 예

- <134>      도 14는 상기 도 6의 복호기 600과 복호기 605의 구조를 나타낸다.
- <135>      도 14를 참조하여 설명하면, 부호화기로부터 부호화된 수신 심볼은 복호기를 구성하는 0 삽입기 1400에 입력된다. 이와 동시에 상기 수신 심볼을 부호화하기 위해 사용된 부호 길이정보가 제어기 1430에 입력되어진다. 상기 제어기 1430은 상기 부호화기에서 사용될 수 있는 모든 부호길이들에 따른 천공위치들을 저장하고 있다. 따라서, 상기 제어기 1430은 상기 수신 심볼에 사용된 부호 길이가 입력되면 상기 입력된 부호길이에 대응하여 저장되어 있는 제어 정보를 상기 0 삽입기 1400으로 출력한다. 예컨대, 상기 제어기 1430으로부터 출력되는 제어정보는 상기 수신 심볼이 어떠한 부호길이를 가지더라도 상기 0 삽입기 1400으로부터 출력되는 심볼의 개수(32 심볼)가 일정한 심볼열을 출력하도록 함을 기준으로 하고 있다. 따라서, 상기 제어기 1430은 부호율이 (3,1)이면, 29개의 천공위치에 대한 정보를, 부호율이 (6,2)이면, 26개의 천공위치에 대한 정보를, 부호율이 (9,3)이면, 23개의 천공위치에 대한 정보를, 부호율이 (12,4)이면, 20개의 천공위치에 대한 정보를, 부호율이 (18,6)이면, 14개의 천공위치에 대한 정보를, 부호율이 (21,7)이면, 11개의 천공위치에 대한 정보를, 부호율이 (24,8)이면, 8개의 천공위치에 대한 정보를, 부호율이 (27,9)이면, 5개의 천공위치에 대한 정보를 출력한다. 각각의 경우에 대하여 천공위치는 상기 부호기 설명에서 제시한 바

와 같다. 그러면 상기 0 삽입기 1400은 상기 천공위치에 대한 제어정보에 따라서 천공위치에 0을 삽입하여 길이 32인 심볼열을 출력한다. 그러면, 상기 심볼열은 역하다마드 변환기 1420과 승산기 1402, 1404, 1406에 각각 입력되어진다. 상기 승산기 1420, 1404, 1406으로 입력되어진 신호들은 마스크 생성기 1410으로부터 생성되어진 모든 경우의 마스크 함수들이 출력되어져 각각 승산되어진 후 출력되어지는데, 상기 출력되어진 심볼들은 스위치 1452, 1454, 1456으로 각각 입력되어진다. 그러면 이와 동시에, 상기 제어기 1430는 상기 입력되어진 부호길이 정보에 따른 마스크함수의 사용여부에 대한 제어정보를 스위치 1452, 1454, 1456으로 각각 출력하는데, (3,1), (6,2), (9,3), (12,4), (18,6)부호기는 마스크 함수를 사용하지 않기 때문에, 상기 제어정보에 따라 스위치 1452, 1454, 1456은 전부 연결을 끊게 된다. 그리고, (21,7)부호기의 경우는 기저 마스크 함수를 1개만 쓰기 때문에, 스위치 1452만 연결이 되고, 상기와 같이 부호율에 따라 사용되어지는 마스크 함수 개수에 따라서 스위치가 제어를 한다. 이 때, 도 6의 상기 복호기 600과 복호기 605가 (3,1), (6,2), (9,3), (12,4), (15,5), (18,6)복호기로 동작하는 경우와 같이 입력되어지는 수신신호가 18보다 작을 때는 단지 역하다마드 변환기 1420만이 동작할 뿐만 아니라, 상기 역하다마드 변환기로 동작하여야 하는데 이와 같은 길이의 월시부호기에 대한 역하다마드 변환기로 동작해야 하는지에 대한 제어신호가 상기 제어기 1430으로부터 생성되어 상기 변환기 1420으로 입력되어진다. 그러면, 상기 역하다마드 변환기 1420, 1424, 1426은 상기 각각 입력된 32개의 심볼을 역하다마드 변환하여, 상기 입력신호에 승산되어

진 마스크 함수의 색인을 나타내는 0 (상기 역하다마드 변환기 1420으로 입력되어진 신호는 어떤 마스크 함수도 승산되어지지 않았기 때문에)과 길이 월시부호와의 상관도 중 가장 높은 상관도를 가지는 월시부호의 색인과 상기 상관도를 계산하여 상관도 비교기 1440에 입력한다. 그러면 상기 1440은 상기 입력된 상관도들을 비교하여 가장 높은 상관도를 가지는 것에 대한 마스크 색인과 월시부호 색인을 결합하여 출력한다.

<136>        도 15는 도 6의 복호기 600과 복호기 605가 (3,1), (6,2), (9,3), (12,4), (15,5), (18,6)복호기로 동작하는 경우에 실제로 상기 하다마드 변환기 1420만이 동작하고, 여러 가지 길이의 월시부호기에 대한 역하다마드 변환기로 동작하는 과정에 대한 전체 구조를 도시한다. 도 15를 참조하여 상기 여러 가지 길이의 월시부호기에 대한 역하다마드 변환기의 동작을 설명하기로 한다.

<137>        이 때, 상기와 같은 역하다마드 변환기의 연산특성을 이용하여, 도 15와 같은 가변길이에 대해 동작 가능한 개략적인 역하다마드 변환기구조를 설명하기로 한다. 도 15를 참조하여 최대길이  $2^n$ 인 월시부호에 대한 역변환까지 가변적으로 가능한 역하다마드 변환기의 구조를 설명하면,  $2^t$ 개의 입력신호가 입력이 되면, 이와 동시 길이에 대한 제어신호가 모든 스위치 1511, 1512에 입력되어진다. 이 때, 상기 제어신호는 상기 입력신호의 길이  $2^t$ 에 대해서 단계 t까지만 수행하도록 하는 제어신호로 단계 t까지 모든 스위치들이 단계간에 연결되고, 단계 t와 단계  $t+1$ 사이의 스위치는 연결을 끊음과 동시에 단계 t에서의 출력신호를 그대로 출력하도록 하는 제어신호이다. 그러면, 상기 입력신호들은 1501의 단계 1에 입력이 되고, 상기

도 16에서 설명되어진 단계 1에 대한 연산과정이 수행되어진 후  $2^t$ 개의 신호를 출력하면, 상기 출력신호들은 스위치 1511에 입력되어진다. 이 때, t가 1이면, 상기 스위치 1511은 단계 2와 연결되어지지 않고 바로 역하다마드 변환기의 출력 단과 연결되어져, 상기 단계1의 출력신호들은 역하다마드 변환기의 출력신호로 출력되어진다. 또한, t가 1이 아니면, 상기 스위치는 단계 2와 연결되어져 다음 단계인 단계 2로 입력되어져 상기 단계 1의 출력신호들에 대해서 단계 2의 연산을 수행하게된다. 상기와 같은 과정으로 단계 t까지 수행되어진 후 단계t에서의 출력값들은 상기 역하다마드 변환기의 출력신호로 출력되어진다.

<138>      도 17은 상기 각각의 단계k에 대한 하드웨어구조를 도시한다. 도 17을 참조하여 설명하면, 먼저  $2^k$ 개의 입력신호의 첫 심볼이 단계 k에 입력되어지면, 감산기 1700, 가산기 1705와 스위치 1710으로 입력되어지면, 상기 스위치 1710은 입력신호가 인가되는 쪽과 연결되어 상기 입력신호가 메모리 1720으로 입력되도록 한다. 스위치 1715는 메모리 1720으로 연결되어진다. 또한 두 번째 입력 심볼이 단계 k에 입력되어지면, 감산기 1700, 가산기 1705와 스위치 1710으로 입력되어지면, 상기 스위치 1710은 입력신호가 인가되는 쪽과 연결되고 상기 메모리 1720에 입력되어진 상기 첫 번째 입력 심볼은 상기 저장된 메모리에서 다음 메모리로 이동되어 저장되어짐과 동시에 상기 두 번째 입력신호가 메모리 1720으로 입력되어진다. 이와 같은 방식으로  $2^k-1$ 개의 입력신호가 입력되어져 상기 메모리 1720에  $2^k-1$ 개의 입력신호가 입력되어지고, 그 후에  $2^k-1 + 1$ 번째 입력신호가 입력되어지면, 상기  $2^k-1 + 1$ 번째 입력신호는 감산기 1700, 가산기 1705와 스위치 1710으로 입력되어

진다. 그러면, 상기 스위치 1710은 감산기 1700으로 연결이 되어지고, 스위치 1715는 가산기 1705로 연결이 되어진다. 그러면, 상기 메모리 1720으로부터 상기 저장되어진 첫 번째 입력 심볼이 감산기 1700, 가산기 1705와 스위치 1715로 입력되어진다. 이 때, 상기 메모리 1720으로부터 상기 첫 번째 입력심볼이 출력됨과 동시에 상기 메모리 1720에 저장되어진 심볼들이 왼쪽으로 한 칸씩 이동하여 저장되어지고, 상기 감산기 1700은 메모리로부터 입력되어진 첫 번째 입력 심볼에서 상기 새로이 입력되어진  $2^{k-1} + 1$ 번째 입력신호를 감산하여 스위치 1710을 통하여 맨 오른쪽 메모리 1720으로 입력되어지고, 이와 동시에 상기 가산기 1705는 상기 메모리로부터 입력되어진 첫 번째 입력 심볼과 상기 새로이 입력되어진  $2^{k-1} + 1$ 번째 입력신호를 가산하여 스위치 1715를 통하여 출력되어진다.

상기와 같은 방법으로  $2^{k-1}$ 번 더 수행되어진 후,  $2^k + 1$ 번째 입력신호가 입력되어지면, 상기 새로이 입력되어진  $2^k + 1$ 번째 입력신호는 감산기 1700, 가산기 1705와 스위치 1710으로 입력되어진다. 이와 동시에, 상기 스위치 1715는 상기 메모리 1720과 연결되어지고, 상기 스위치 1710은 입력신호단과 연결되어지고, 상기 메모리 1720으로부터 맨 왼쪽에 입력되어진 첫 번째 입력 심볼에서  $2^{k-1} + 1$ 번째 입력신호를 감산한 값은 스위치 1715를 통해 출력되어지고, 이와 동시에, 상기 1720에 저장되어져있는 값들은 왼쪽으로 한 칸씩 이동되어 저장되어지고, 이와 동시에 상기 새로이 입력되어진  $2^k + 1$ 번째 입력신호는 스위치 1710을 통하여 상기 메모리 1720의 맨 오른쪽으로 입력되어진다. 상기와 같은 과정이 반복되어져 입력신호의 길이 만큼이 출력이 되어지면, 상기의 단계 $k$  과정이 모두 수행되어진 것이다.

<139> 본 발명의 제4 및 제5 목적을 달성하기 위한 방법을 도 10, 11, 12, 13을 설명한다.

<140> 도10는 상기 Logical split 방법을 위한 기지국간의 신호메시지 및 데이터 전송을 나타내는 도면이다.

<141> 상기 도 10의 단계 301에서 SRNC의 RLC는 SRNC의 Mac-d에 전송할 DSCH data를 전송한다.

<142> 상기 도 10의 단계 302에서 SRNC의 Mac-d는 DRNC의 Mac-c에 RLC로부터 수신한 DSCH data를 전송한다. 이 때 전송되는 data는 Iur상의 Frame protocol을 이용하여 전송된다.

<143> 상기 도 10의 단계 303에서는 DRNC의 Mac-c는 단계 302에서 SRNC의 Mac-d에서 수신한 DSCH data에 대하여 전송시간을 결정한 후 (scheduling) 해당 TFCI를 SRNC의 Mac-d로 전송한다.

<144> 상기 도 10의 단계 304에서 DRNC의 Mac-c는 Node B의 L1로 DSCH 데이터를 전송한다. 이 때 전송되는 DSCH 데이터는 단계 303에서 미리 결정된 (scheduled) 시간에 전송된다.

<145> 상기 도 10의 단계 305에서 SRNC의 Mac-d는 Node B의 L1으로 DSCH를 위한 TFCI와 전송시간을 함께 전송한다. 이 때 전송되는 data는 control frame을 이용하여 전송된다.

<146> 상기 도 10의 단계 306에서 SRNC의 Mac-d는 Node B의 L1으로 전송될 DCH의 data와 DCH를 위한 TFCI를 결정하여 전송한다.

- <147> 상기 도 10의 단계 304에서 전송되는 DSCH data와 단계 305에서 전송되는 TFCI는 상기 단계 303에서 결정된 시간과 관계를 갖는다. 즉 단계 305에서 전송되는 TFCI는 단계 304에서 전송되는 DSCH 데이터가 PDSCH로 전송되기 직전 Frame에 DPCCH로 UE에게 전송되어 진다.
- <148> 상기 도 10의 단계 304와 단계 305와 단계 306에서 전송되는 데이터 및 TFCI는 Frame protocol을 이용하여 전송된다. 특히 단계 306에서 전송되는 TFCI는 Control frame을 통하여 전송된다.
- <149> 상기 도 10의 단계 307에서 Node B의 L1은 UE의 L1으로 DSCH data를 PDSCH로 전송한다.
- <150> 상기 도 10의 단계 308에서 Node B의 L1은 UE의 L1으로 DPCH를 이용하여 TFCI를 전송한다. 이때 상기 단계 305과 단계 306에서 수신한 각각의 TFCI 또는 TFI를 이용하여 하나의 TFCI를 생성하여 DPCCH를 이용하여 전송한다.
- <151> 도 11은 SRNC의 동작과정을 나타내는 도면이다.
- <152> 상기 도 11의 단계 401에서 SRNC는 전송할 DSCH data를 준비한다.
- <153> 상기 도 11의 단계 402에서 SRNC는 준비된 DSCH data를 DRNC로 전송한다.
- <154> 상기 도 11의 단계 403에서 SRNC는 DRNC로부터 상기 단계 402에서 전송한 DSCH data에 대한 Scheduling 정보를 수신한다. 이 때 수신되는 정보는 Control Frame을 이용하여 전송될 수 있으며 정보의 내용으로는 전송시간과 TFCI에 관한 정보가 될 수 있다.

- <155> 도 13은 이 때 DRNC로부터 SRNC로 전송되는 정보를 담고 있는 Control Frame의 구조의 한 예를 보여 주는 도면이다.
- <156> 상기 도 13에서 CFN은 Connection Frame Number로써 전송될 Frame의 번호를 나타내며 이것은 DSCH가 전송될 시간에 대한 정보이다. 상기 도 8에서 TFCI(field 2)는 전송될 DSCH에 대한 TFCI 정보이다.
- <157> 상기 도 11의 단계 404에서 SRNC는 Node B로 해당 DSCH에 대한 전송시간과 TFCI 정보를 담은 Control Frame을 전송한다. 이 때 Control Frame은 해당 전송 시간 전에 Node B에 도착해야 한다.
- <158> 도 12은 DRNC의 동작과정을 나타내는 도면이다.
- <159> 상기 도 12의 단계 501에서 DRNC는 SRNC로부터 DSCH data를 수신한다.
- <160> 상기 도 12의 단계 502에서 DRNC는 상기 단계 501에서 수신한 DSCH data를 전송한 Scheduling을 실시한다. DRNC는 여러 RNC로부터 수신한 DSCH와 DRNC 자체에서 생성된 DSCH들을 전송할 시간들을 결정하고 또한 전송시 사용할 Channel등을 고려한 TFI 또는 TFCI를 결정한다.
- <161> 상기 도 12의 단계 503에서 DRNC는 상기 단계 502에서 결정된 시간정보와 TFCI 정보를 control frame을 이용하여 SRNC에 전송한다. 이 때 전송되는 Control frame의 구조는 상기 도 8에 나타나 있다.
- <162> 상기 도 12의 단계 504에서 DRNC는 상기 단계 502에서 결정된 시간에 Node B에서 전송될 수 있도록 DSCH data를 Node B에 전송한다.

### 【발명의 효과】

<163> 상술한 바와 같이 본 발명의 실시예에서는 TFCI를 부호화 및 복호화할 때, 하나의 코더/디코더 구조를 이용하여 다양한 종류의 TFCI 비트들을 부호화/복호화할 수 있다. 또한 서로 다른 코딩 방법을 사용하여 부호화된 복수의 TFCI 심볼들을 전송할 때, 상기 TFCI 심볼들이 균일하게 분포되어 전송될 수 있도록 다중화할 수 있다. 여기서 TFCI 코딩은 10비트인 경우, DSCH 및 DCH 채널들의 데이터 전송량에 따라 1:9, 2:8, 3:7, 4:6, 5:5, 6:4, 7:3, 8:2 또는 9:1 중의 어느 하나를 선택하여 사용할 수 있다. 또한 본 발명의 실시예에 따르면 로지컬 스플리트 방법의 경우 SRNC와 DRNC가 분리되어 있으면 스케줄링 정보를 Mac-c에서 Mac-d로 전송할 수 있으며, 또한 DSCH 의 두 개의 TFCI 송신 방법인 하드 스플리트 방법과 로지컬 스플리트 방법을 구분하려 사용할 수 있도록 신호 메시지를 전송할 수도 있다.

**【특허청구범위】****【청구항 1】**

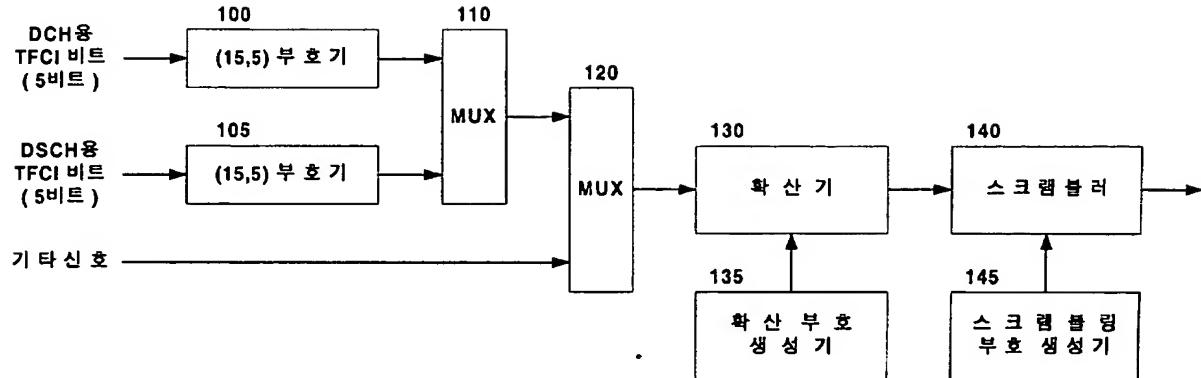
전송하고자 하는 데이터의 부호율에 의해 월시부호의 길이를 가변하여 사용하는 부호기로부터의 데이터를 수신하고, 최대 길이의 월시부호에 의해 부호화된 데이터에 대해서도 복호화가 가능한 최대 역하다마드 변환 단계들을 가지는 복호기에서 상기 수신한 데이터를 복호하는 방법에 있어서,

상기 수신한 데이터에 사용된 월시부호의 길이에 의해 상기 최대 역하다마드 변환 단계들 중 사용할 적어도 하나의 역하다마드 변환 단계를 결정하는 과정과,

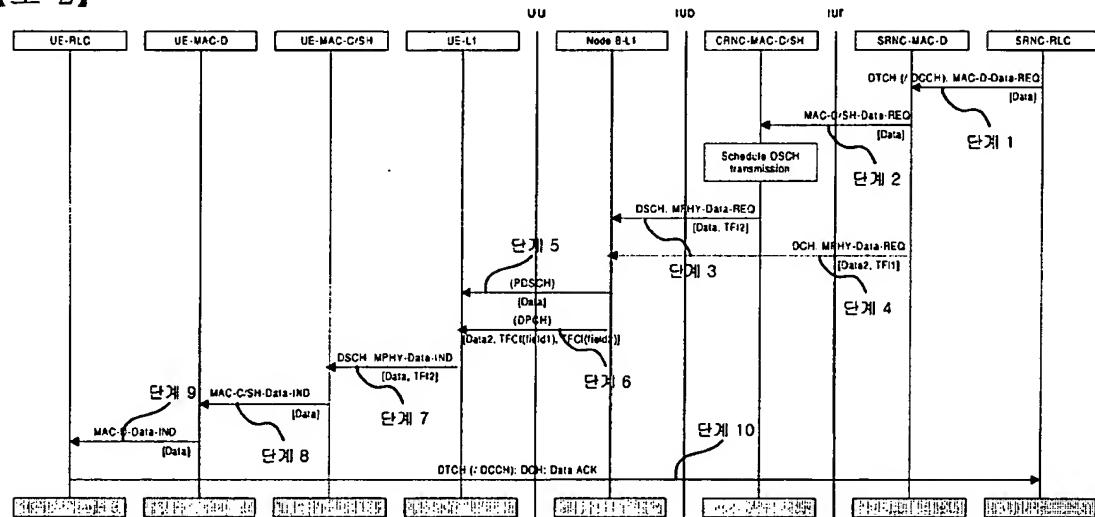
상기 수신한 데이터를 상기 결정한 적어도 하나의 역하다마드 변환 단계에 의한 역하다마드 변환을 수행하는 과정을 포함함을 특징으로 하는 상기 방법.

## 【도면】

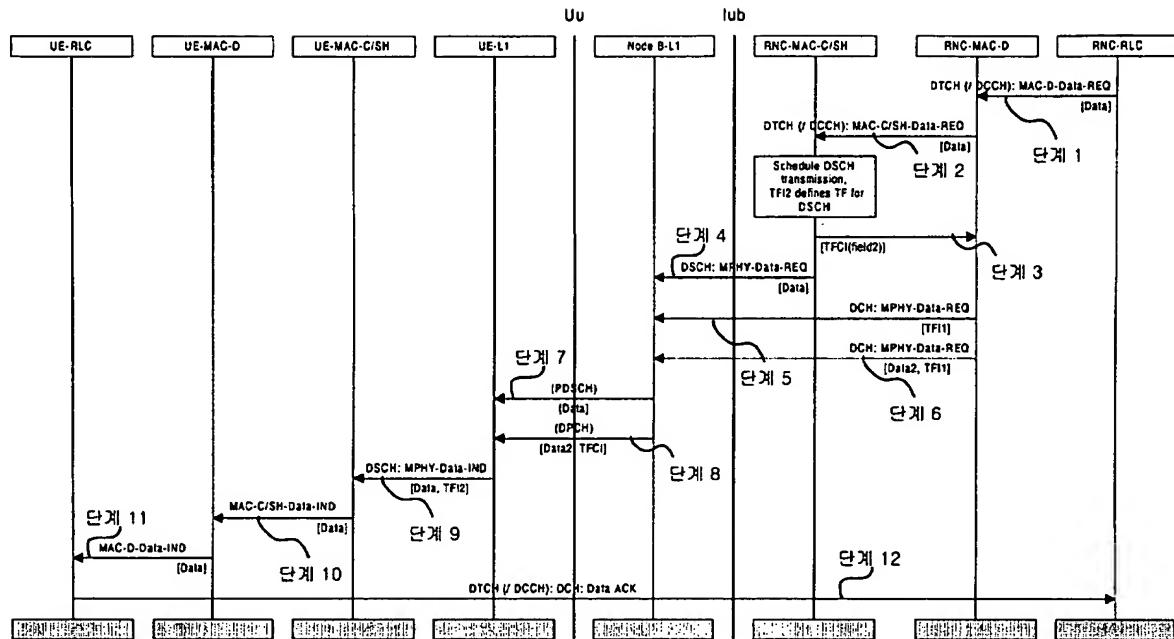
【도 1】



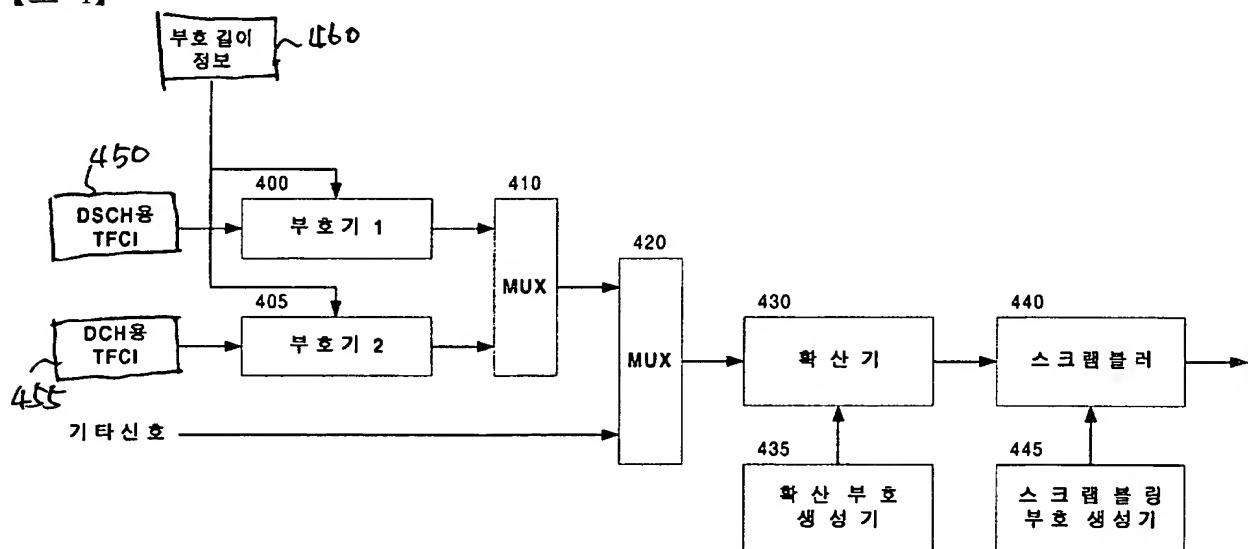
【도 2】



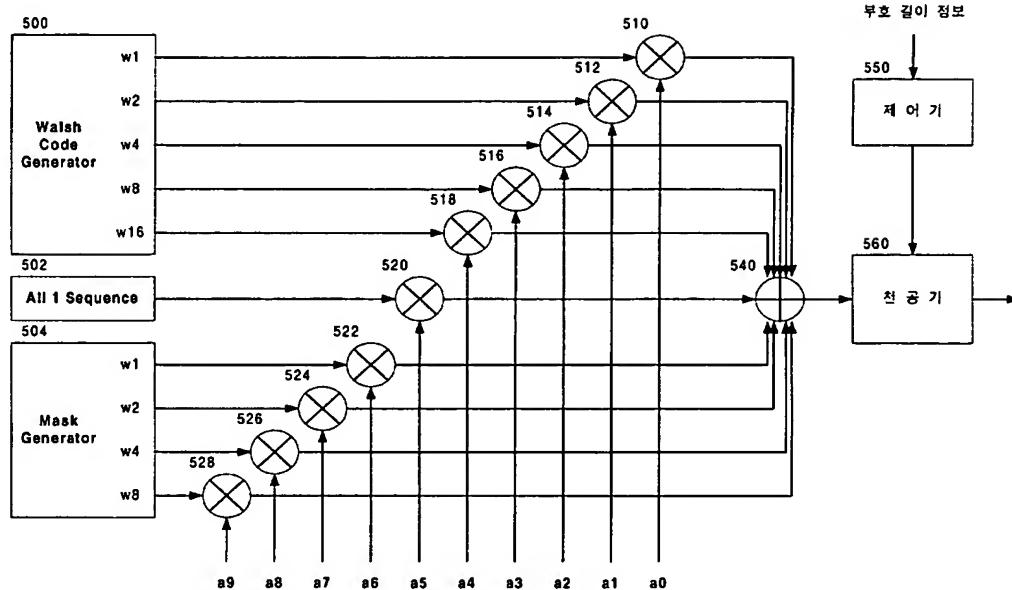
## 【도 3】



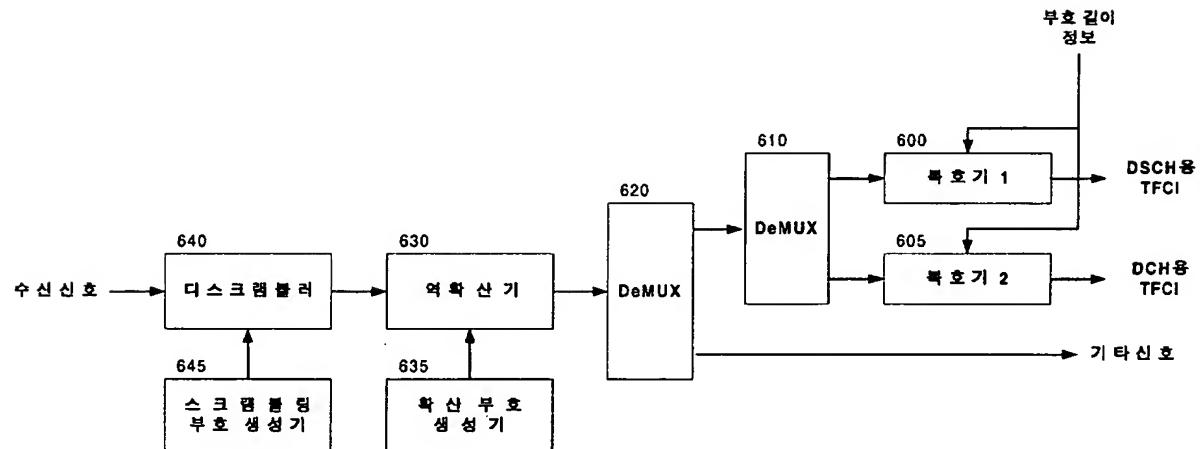
## 【도 4】



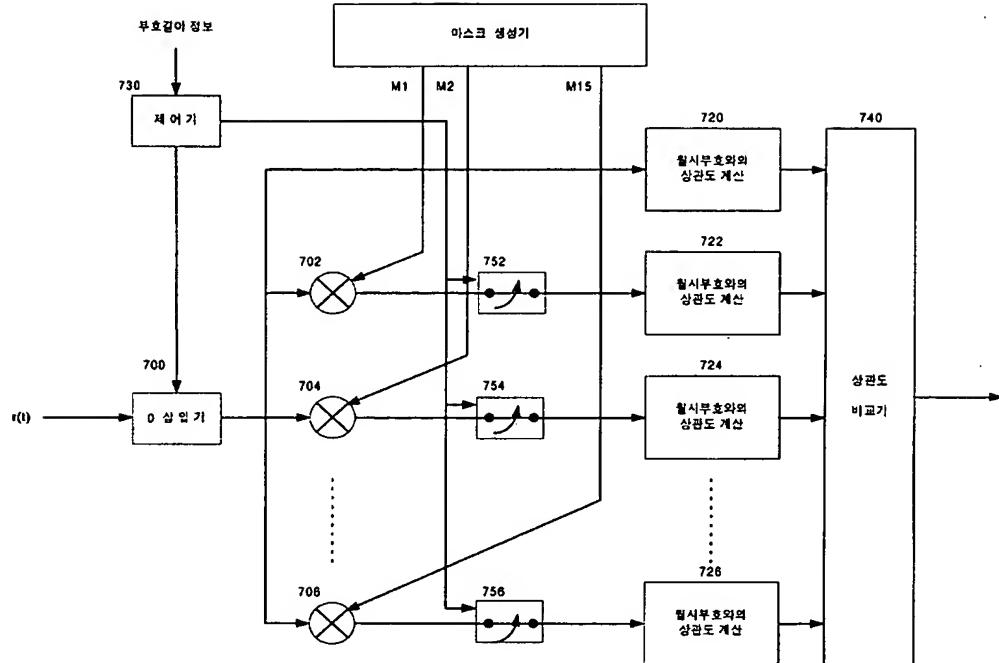
【도 5】



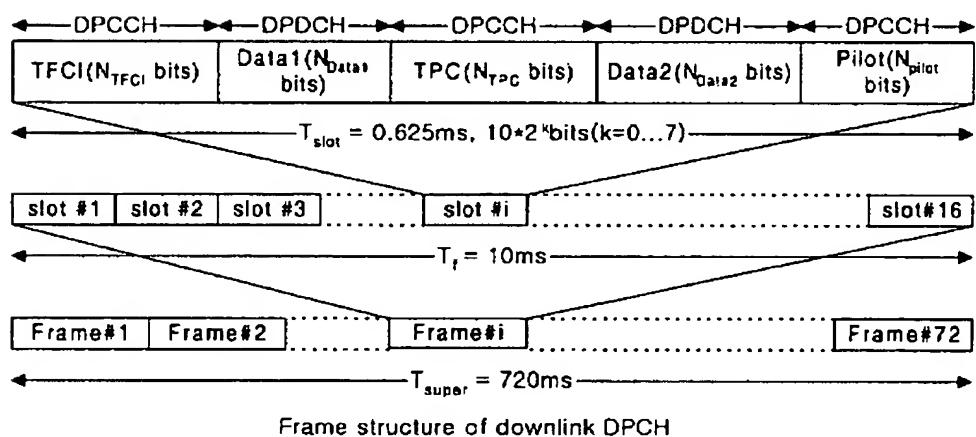
【도 6】



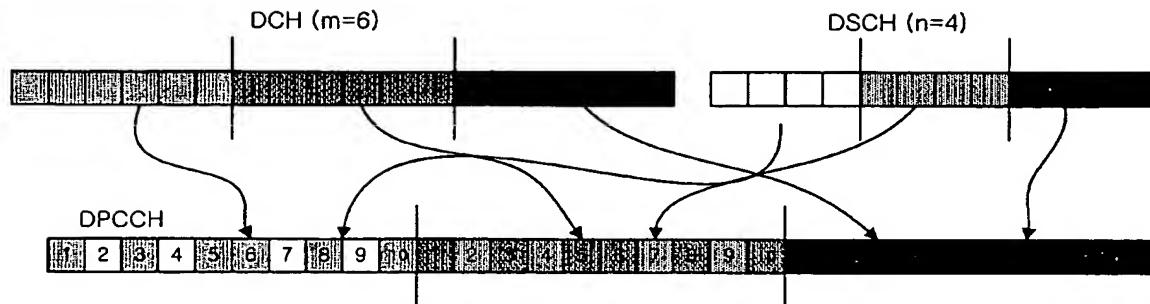
【도 7】



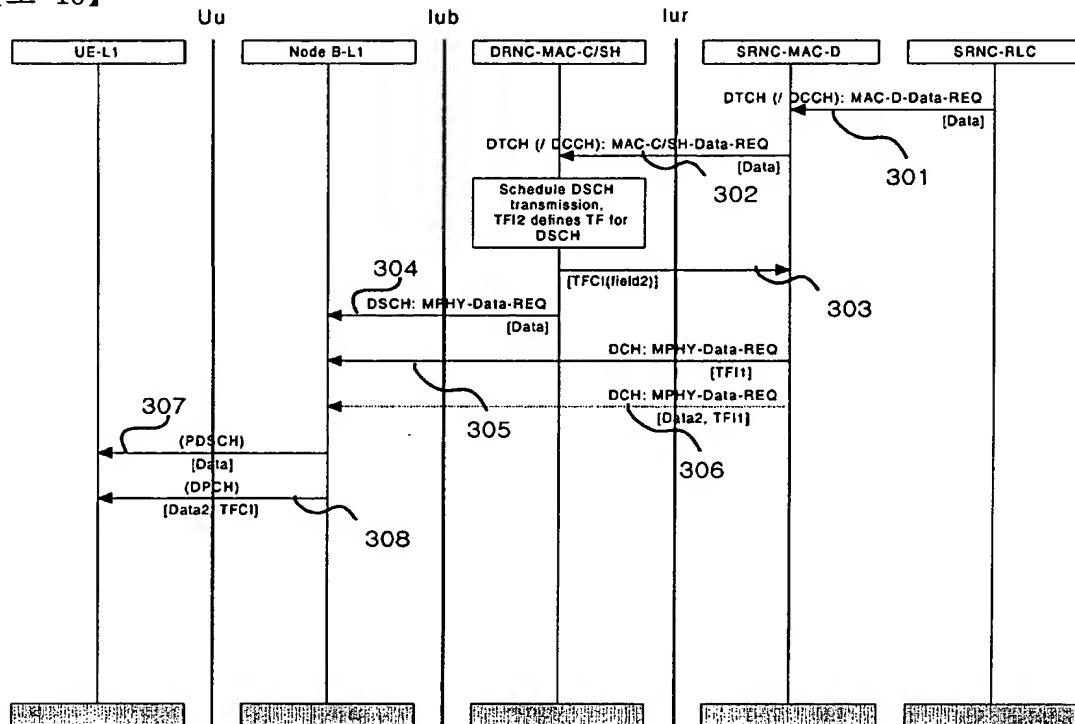
【도 8】



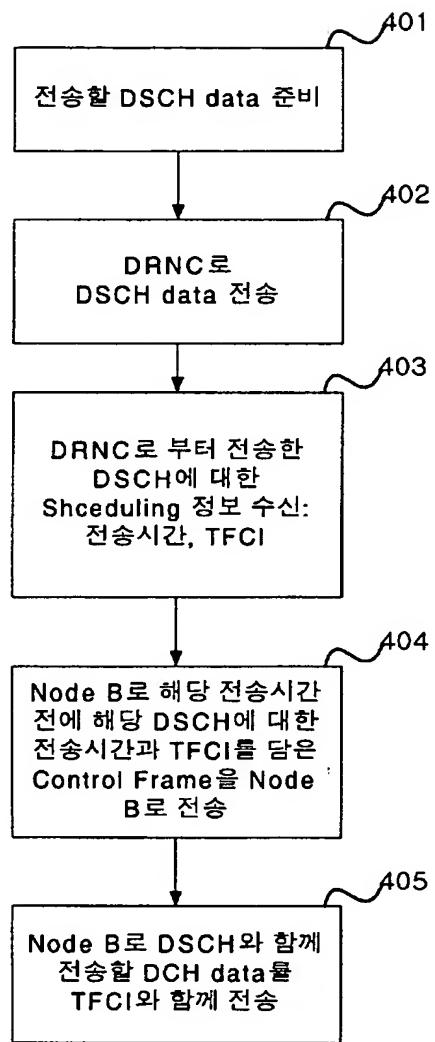
【도 9】



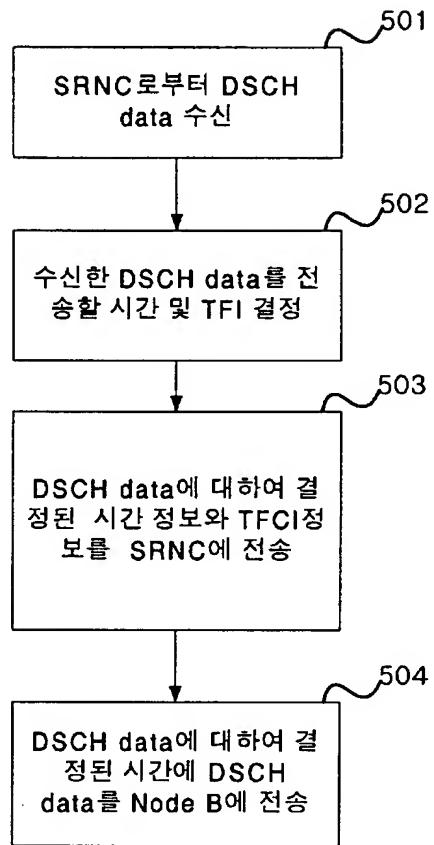
【도 10】



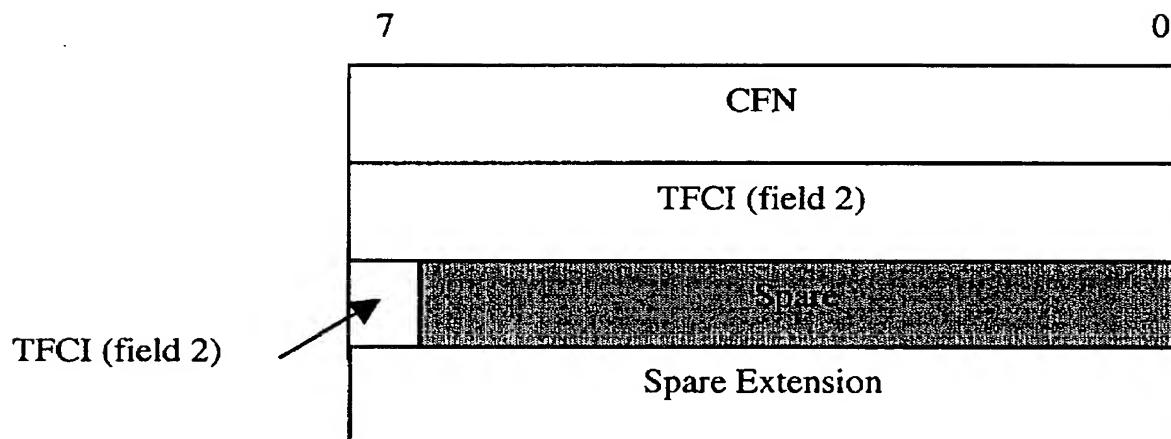
【도 11】



【도 12】



【도 13】

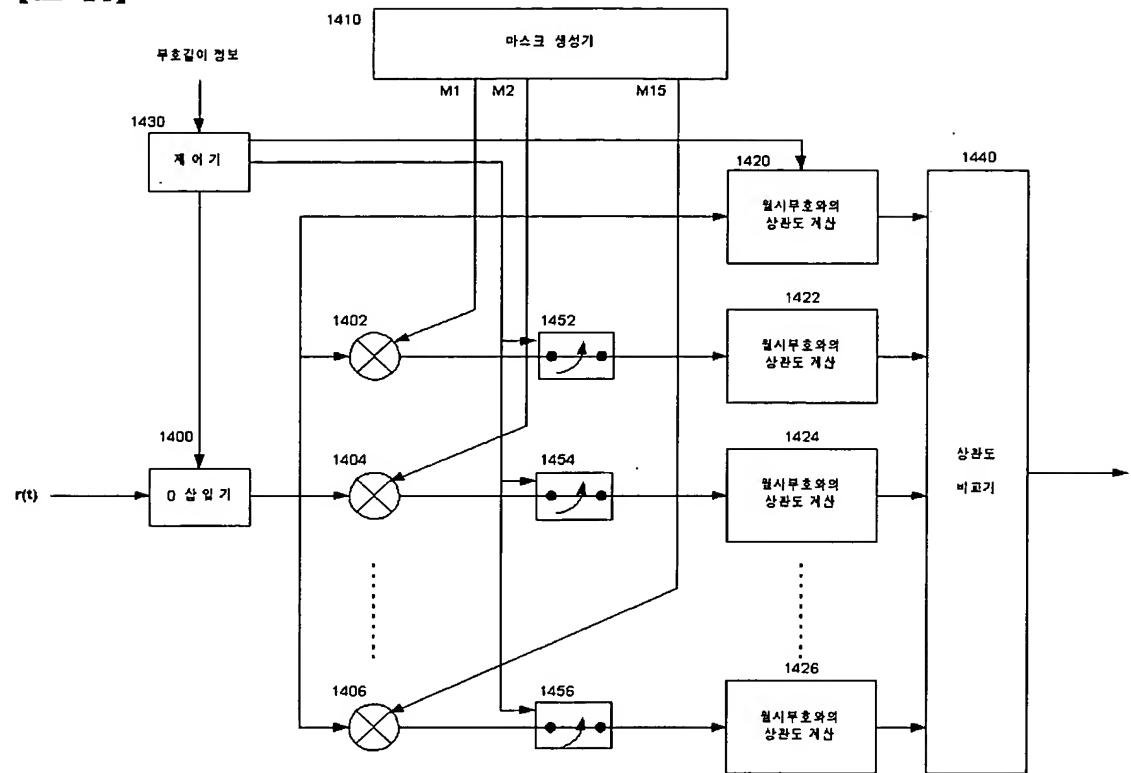




1020010010150

출력 일자: 2002/2/28

【도 14】





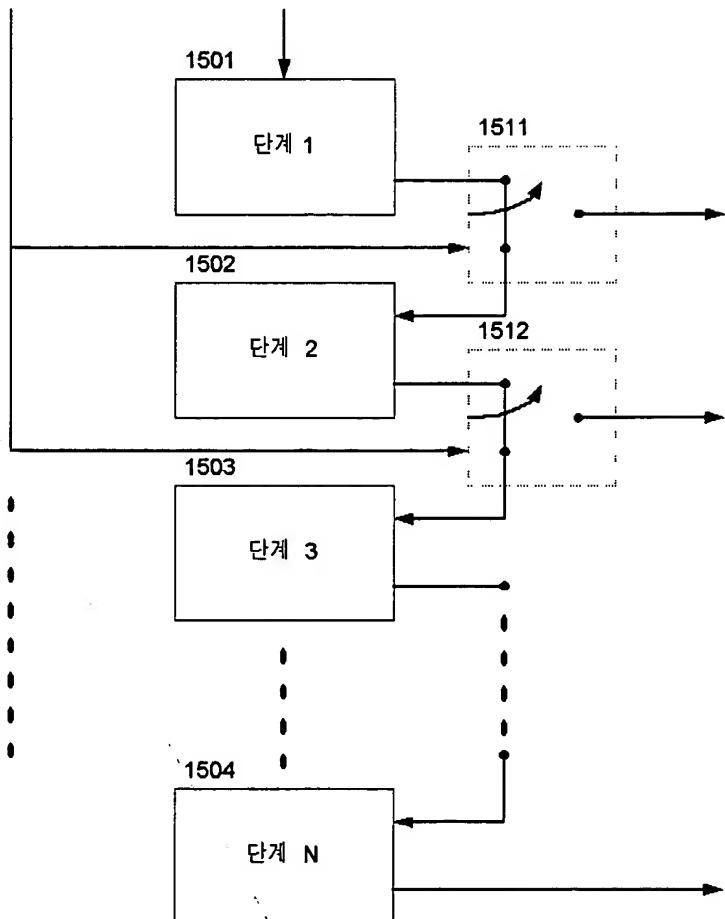
1020010010150

출력 일자: 2002/2/28

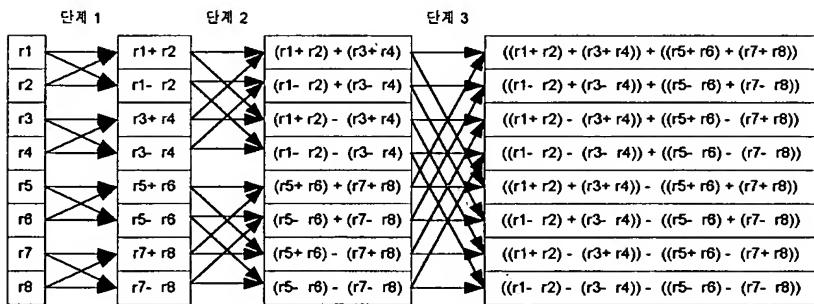
【도 15】

## 제어신호

## 2개의 입력신호



【도 16】





1020010010150

출력 일자: 2002/2/28

【도 17】

